Saklı Markov Modelleri ve Sürekli Konuşma Tanıma Tekniğiyle Rakam Dizisi Tanıma

Alper Tunca

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Temmuz 2010

Digit Sequence Recognition Using Hidden Markov Models and Continuous Speech Recognition Technique

Alper Tunca

MASTER OF SCIENCE THESIS

Department of Electrical-Electronics Engineering

July 2010

Saklı Markov Modelleri ve Sürekli Konuşma Tanıma Tekniğiyle Rakam Dizisi Tanıma

Alper Tunca

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Telekomünikasyon Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Yrd. Doç. Dr. Rifat EDİZKAN

Temmuz 2010

ONAY

Elektrik-Elektronik Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi Alper Tunca'nın YÜKSEK LİSANS tezi olarak hazırladığı "Saklı Markov Modelleri ve Sürekli Konuşma Tanıma Tekniğiyle Rakam Dizisi Tanıma" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Danışman **:** Yrd. Doç. Dr. Rifat EDİZKAN

İkinci Danışman **: -**

Yüksek Lisans Tez Savunma Jürisi:

Üye : Yrd. Doç. Dr. Rifat EDİZKAN

Üye : Prof. Dr. Atalay BARKANA

Üye : Prof. Dr. M.Bilginer GÜLMEZOĞLU

Üye : Yrd. Doç. Dr. Erol SEKE

Üye : Yrd. Doç. Dr. Kemal ÖZKAN

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

ÖZET

Son yıllarda sürekli konuşma tanıma konusunda büyük ilerlemeler kaydedilmiştir. Birçok alanda uygulamalar geliştirilmiştir. Türkçe için de bazı uygulamalar ve çalışmalar yapılmıştır. Türkçe'nin, sondan eklemeli bir dil olarak, dağarcık boyutu çok fazladır. Dağarcık boyutu yüksek bir uygulama geliştirmek için fonem tabanlı bir konuşma tanıma sisteminin geliştirilmesi gerekir. İleride yapılacak karışık ve kapsamlı uygulamalara bir kaynak ve altyapı teşkil etmesi amacıyla bu tez kapsamında fonem tabanlı bir sürekli konuşma tanıma sisteminin geliştirilmesi yapılmıştır. Bu sistem üzerinde üç basamaklı sayı ve dörtlü rakam dizisinin tanınması uygulaması geliştirilmiştir. Bu uygulama not girişi, haberleşme sistemlerinde frekans girişi, otomatik telefon numarası çevirimi gibi gerçek uygulamalar için bir temel oluşturabilir. Rakam dizisi tanıma uygulaması için gerekli tüm fonemler için, kelime içindeki durumlarına göre üçlü fonem (trifon) bazında Saklı Markov Modelleri (Hidden Markov Models) oluşturulmuş ve eğitilmiştir. Eğitim sonucu elde edilen SMM'ler ile rakam dizisi tanıma uygulamaları yapılmış ve elde edilen başarımlar değerlendirilmiştir.

Anahtar Kelimeler: Sürekli konuşma tanıma, Saklı Markov Modeli.

SUMMARY

Recently, there are many improvements in continuous speech recognition systems. Several applications have been developed in various fields. Also many practical applications have been developed in Turkish. Turkish, as an agglunative language, it has a vast vocabulary. Phoneme based training should be performed for large vocabulary speech recognition systems. In this thesis, in order to assist and to be background for more complex and comprehensive applications, an application about digit sequence recognition was developed. This application can be utilized in many practical applications like grade entry, frequency selection in communication systems, phone number dialing. For digit sequence recognition application, Hidden Markov Models were developed and trained for phonemes in Turkish, as context dependent triphones. At the end of training several recognition tests are performed and results were evaluated.

Keywords: Continuos Speech Recognition, Hidden Markov Model, Hidden Markov Tool Kit

TEŞEKKÜR

Yüksek Lisans tez çalışmalarımda, bana danışmanlık ederek, beni yönlendiren ve her türlü olanağı sağlayan danışmanım Yrd. Doç. Dr. Rifat EDİZKAN'a ayrıca katkılarından dolayı tüm çalışma arkadaşlarıma ve bu dönemde bana gösterdiği sabır ve desteklerinden dolayı sevgili eşim Tuba'ya ve kızım Zeynep Yağmur'a sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER (devam)

Sayfa

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

KISALTMALAR

Kısaltma	Açıklama
ASR	Automatic Speech Recognition
DCT	Discrete Cosine Transform
DTW	Dynamic Time Warping
HTK	Hidden Markov Tool Kit
FFT	Fast Fourier Transform
IPA	International Phonetic Alphabet
LPC	Linear Predictive Coefficient
LVCSR	Large Vocabulary Continuos Speech Recognition (Geniş Dağarcıklı Sürekli Konuşma Tanıma)
METUbet	Middle East Technical University Alphabet
MFCC	Mel Frequency Cepstral Coefficient (Mel Frekans Dağılım Katsayıları)
SAMPA	Speech Assessment Method Phonetic Alphabet
SMM	Saklı Markov Modeli (Hidden Markov Model)

BÖLÜM 1

GİRİŞ

Konuşma, insanların birbirleri ile iletişim için ses telleri, gırtlak, dil, dudak gibi organların beraber koordineli çalışarak ürettikleri anlamlı sesli mesajlara denir. Hayvanların kısıtlı bazı mesajlar iletmek için kullandığı farklı seslere kıyasla, insanoğlunun hayal gücü ve ihtiyaçları ile paralel olarak konuşma çok daha karmaşıktır ve çok daha kapsamlı mesajlar iletir.

İlk olarak tüm insan toplulukları belki tek bir dille konuşurken, değişik coğrafyalara yayılarak, birbirlerinden uzaklaşarak değişik kuralları ve dağarcıkları olan dilleri oluşturmuşlardır. İlk önce belki çok basit olan, doğadaki sesleri taklit olan diller, yüzyıllarca süren bir süreç içinde belli bir gramer yapısına, geniş bir sözcük dağarcığına kavuşmuştur. Diller hala gelişmeye devam etmektedir. Teknolojik ilerleme ile dile hergün yeni terimler girmektedir.

Bilgisayarlar geliştikçe insanların bilgisayarlı sistemlerle konuşarak iletişim kurma gerekliliği ortaya çıkmıştır. Klasik klavye veya fare gibi yöntemler yerine konuşmayı anlayan akıllı sistemler geliştirmek için çalışmalar yapılmış olup, bu konu üzerindeki ilk çalışmalar 1950'lere dayanır. (Rabiner and Juang, 1993).

Ses tanıma sisteminin geliştirilmesi zorlu bir süreçtir. İlk çalışmalar İngilizce dilinde yapılmıştır. Her dil için farklı ses tanıma yazılımları oluşturulması gerekir. Her dilin özelliklerine göre sorunların ele alınması, çözüm bulunması gerekir. Sondan eklemeli bir dil olan Türkçe için geniş dağarcıklı bir konuşma tanıma sistemi geliştirmek zordur. Türkçe üzerine de üniversitelerimizde değişik çalışmalar yapılmaktadır. Hatta bu konuda çalışan ticari firmalar da vardır (örneğin SESTEK, 2010 ve DİCTE, 2010).

Konuşma tanıma sistemleri tanıma yetenekleri yönüyle birbirinden ayrılırlar. Bu ayrım bir konuşmacının kullandığı sözcüklerin başlangıç ve bitiş yerlerinin tespitine göre değişir. Temel sistemler yalıtık konuşma tanıma, bağlantılı konuşma tanıma ve sürekli konuşma tanımadır. Yalıtık konuşma tanımada her sözcük arasında duraklama

gerekmektedir. En basit sistemdir ve uç belirleme teknikleriyle kelimeler teker teker tanınır. Bu şekilde kelime tanıma daha kolay yapılır. Bağlantılı konuşma tanıma ise yalıtık konuşma tanımaya benzer. Kelimeler arasında duraklama olması gerekli değildir. Bağlantılı konuşma tanıma sisteminde gramer yapısı oluşturularak dizi veya cümle tanınabilir. Gramer üzerinden hangi dizi veya cümlenin söylendiğini belirleme süresi uzundur. Sürekli konuşma tanıma ise birçok yönden zordur. Bu tanıma şekli büyük kelime kütüphanesine sahip uygulamalarda kullanılır. Kelimler fonetik sesbirimleri kullanılarak modellenir. Bu şekilde yüksek kelime tanınması istenilen uygulamalar gerçeklenebilir. Burada kelimler doğal olarak söylenir ve bu nedenle aralarında duraklama yoktur. Burada kelime sınırlarının belirli olmaması sürekli konuşma tanıma işlemi için önemli bir problemdir. Sürekli konuşma tanımada sözcük sınırlarının tespiti için değişik metotlar kullanılması gerekir. Sözcük içindeki her fonem, önündeki ve arkasındaki fonemlere göre değişim gösterir. Benzer şekilde sözcükler de cümle içinde önündeki ve arkasındaki sözcüklere göre değişim gösterir. Ayrıca konuşmanın hızı da tanımayı zorlaştırır. Tezin düzeni şu şekildedir:

Bölüm 2'de sürekli konuşma tanıma sistemi tanıtılmakta, daha sonra bölüm 3'de HTK sistemi tanıtılmakta, bölüm 4'de Türkçe'nin ses yapısı, bölüm 5'de ise yapılan uygulama hakkında bilgi verilmektedir. Son olarak bölüm 6'da yer alan sonuç ve öneriler kısmı ile aldığımız sonuçlar ve değerlendirmeler sunulmaktadır.

BÖLÜM 2

KONUŞMA TANIMA

Konuşma tanıma sistemleri, ses sinyalini oluşturan sıralı sembol dizinini çözmeyi amaçlar. Ana amaç, iletilen mesajı çözme,, daha sonra yazıya veya işlem yapmak üzere komutlara çevirmektir.

Konuşma tanımaya üç farklı yaklaşım vardır (Rabiner and Juang, 1993).

1. Akustik-fonetik yaklaşım
2. Örüntü (pattern) tanıma yaklaşımı
3. Yapay zeka yaklaşımı.

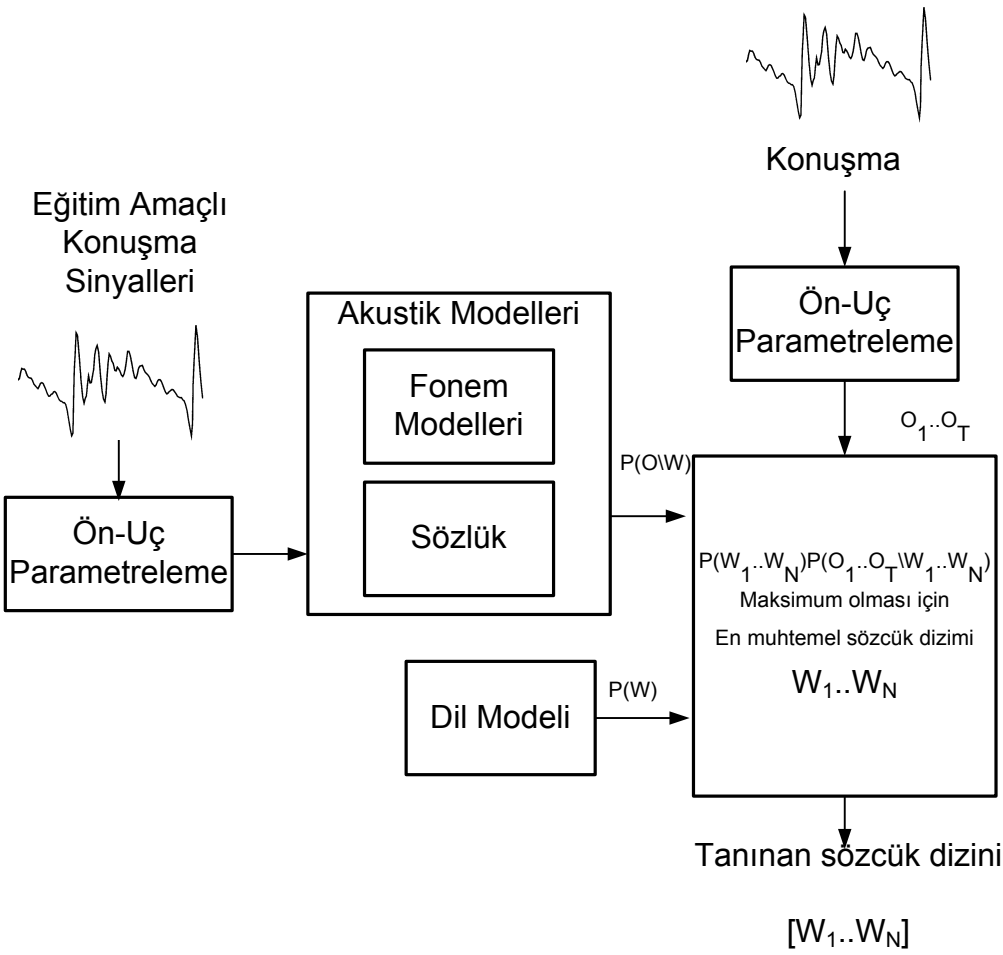
Akustik fonetik yaklaşımda, konuşulan dilde ayrı fonetik öğelerin var olduğu düşünülür. Fonetik öğe, sahip olduğu birçok özelliğe göre birbirlerinden ayrılır.

Örüntü tanıma yaklaşımında, iki basamaklı bir yapı vardır: İlk olarak konuşma örüntülerinin eğitimi ve daha sonra test örüntülerinin kıyaslama yoluyla tanınması. Sisteme, eğitim yoluyla konuşma bilgisi sağlanır. Mevcut ASR sistemleri istatistiksel örüntü tanıma metoduyla çalışır.

Yapay zeka yaklaşımı ise, ilk iki metodu birleştirir. Bu yaklaşım, insanın akustik bilgisi ve tecrübesiyle zekasını kullanarak konuşmayı tanımasına benzer.

2.1 İstatistiksel Konuşma Tanıma

Konuşma tanıma için istatistiksel örüntü tanıma yaklaşımı en son gelinen teknolojide en sık kullanılan metottur. Bu yaklaşımda, konuşma tanıma problemini, bir ses verisindeki akustik bilgiyi kullanma ve buradan söylenilen kelime dizisini elde etmek olarak tanımlayabiliriz. Konuşma tanıma sistemleri içinde en gelişmişi ve karmaşığı olan sürekli konuşma tanıma sistemi Şekil 2.1'de gösterilmektedir.



Şekil 2.1 Sürekli konuşma sistemi

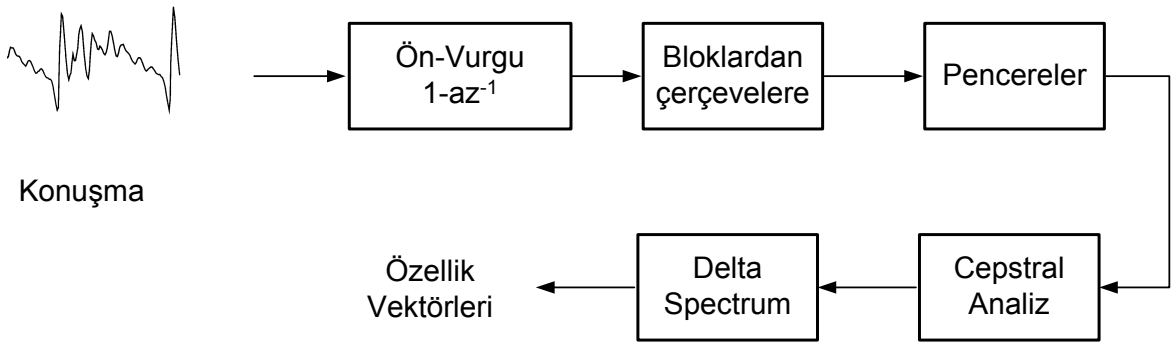
Bilinmeyen konuşma sinyalinin $Y=y_1, y_2, \ldots, y_T$ akustik vektörlerden oluştuğu düşünülebilir. Bunlar konuşma verisinden özellik çıkarma ile elde edilir. Ayrıca, bu akustik vektörler $W=w_1, w_2,.., w_N$ olarak gösterilen belli bir kelime sırasını oluşturmakta ve bunlar üzerinden olasılığı en yüksek kelime dizisi $\hat{W}$ elde edilmektedir. Denklem 2.1 ile en yüksek olasılığı veren kelime dizisi W'yi bulmak için, $P(W)$ ve $P(Y|W)$ çarpımının en büyük yapacak kelime sırası tespit edilmelidir. Bunu yapmak için Bayes kuralı ile $P(W|Y)$ iki birime ayrılır.

$$\hat{W} = \arg_w \max P(W|Y) = \arg_w \max \frac{P(W)P(Y|W)}{P(Y)} \qquad (2.1)$$

İlk terim W'nin gözlenen sinyalden bağımsız olan ve dil modeli ile tespit edilen önsel olasılığıdır. İkinci terim ise belli bir W kelime sırası için Y vektör sırasının görülme olasılığıdır ve akustik model ile tespit edilir.

2.1.1 Ön-uç parametreleme

Ön uç parametrelemenin amacı konuşma sinyalinden özellik vektörleri çıkarmaktır. Özellik vektörleri konuşmacıdan ve ortamdan bağımsız olmalıdır. Bu nedenle, bir filtre olarak görülen insanın ses üreten gırtlak yapısı modellenmeye çalışılmıştır. Bu modeller bu filtrenin parametrelerini tahmini olarak hesaplar. Hesaplama kolaylığı açısından, mevcut tanıyıcı sistemlerde konuşma sinyalinin 10-20 milisaniye boyunca durağan olarak kabul edilmektedir. Ön-uç işlemci blok şeması Şekil 2.2'de görülmektedir.



Şekil 2.2 Ön-Uç işlemci blok şeması

2.1.1.1 Kısa zaman analizi

Konuşmanın kısa zaman aralıkları ile işlenmesine kısa zaman analizi adı verilir. Temel olarak konuşma, bölümlere ayrılır ve bu bölümler birbirinden bağımsız olarak işlenir. Konuşma sinyali *x[n]*'nin m'deki kısa zamanlı durumu aşağıdaki gibidir (Huang et al., 2001). Bu ifadede $w_m[n]$ pencereleme fonksiyonudur.

$$x_m[n] = x[n]w[m-n] \qquad (2.2)$$

Hamming penceresi, konuşma sinyalinin kısa-zamanlı analizinde kullanılır. N boyutunda bir Hamming penceresi Denklem (2.3) ile verilir.

$$h[n] = \begin{cases} 0.54 - 0.46\cos(2\pi \frac{n}{N}) & 0 \le n \le N \\ 0 & diger \end{cases} \quad (2.3)$$

Kısa zamanlı discrete Fourier tanımı aşağıda şekilde tanımlanır (Quatieri, 2002):

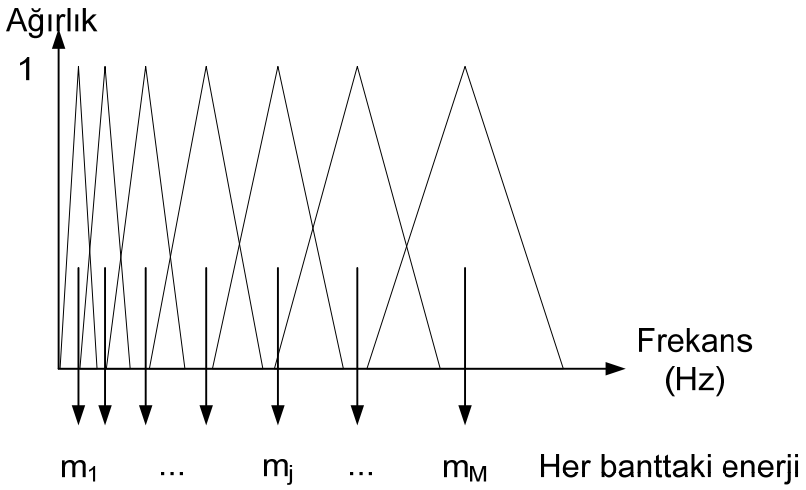
$$X(m,k) = \sum_{n=-\infty}^{\infty} w[m-n]x[n]e^{-j\frac{2\pi}{N}kn} \quad (2.4)$$

2.1.1.2 Mel-Frekans Dağılım Katsayıları

İnsanın kulağı ses sinyalinin temel frekanslarını doğrusal algılayamaz. İnsanın duyma algısı 1 kHz 'e kadar doğrusal, 1 kHz'nin üzerinde ise logaritmiktir. "Mel-frequency cepstral coefficients" (Davis and Mermelstein, 1980) adı verilen Mel-frekans dağılım katsayıları (MFCC) bu algılama düzeyi için geliştirilmiştir.

Ayrık kısa zamanlı Fourier Transform, ses sinyallerinin MFCC'leri çıkarmak için kullanılır. Ayrık kısa-zamanlı Fourier dönüşüm alındıktan sonra, mel-scale adı verilen, insanların algısına yakın doğrusal olmayan frekans boyutuna indirgenir. Denklem 2.5'de mel-scale'in nasıl hesaplandığı gösterilmektedir. Frekans f'nin mel-scale'i *Mel(f)*'dir. İnsan kulağı algılama yapısına uygun olarak ses sinyali Mel ölçüsüne göre yerleştirilmiş M adet filtreden geçirilir. Daha sonra DCT (Discrete Cosine Transform) uygulanarak MFCC parametreleri elde edilir.

$$Mel(f) = 1125\ln(1 + f/700) \quad (2.5)$$



Şekil 2.3 Mel-Scale filtre kümesi

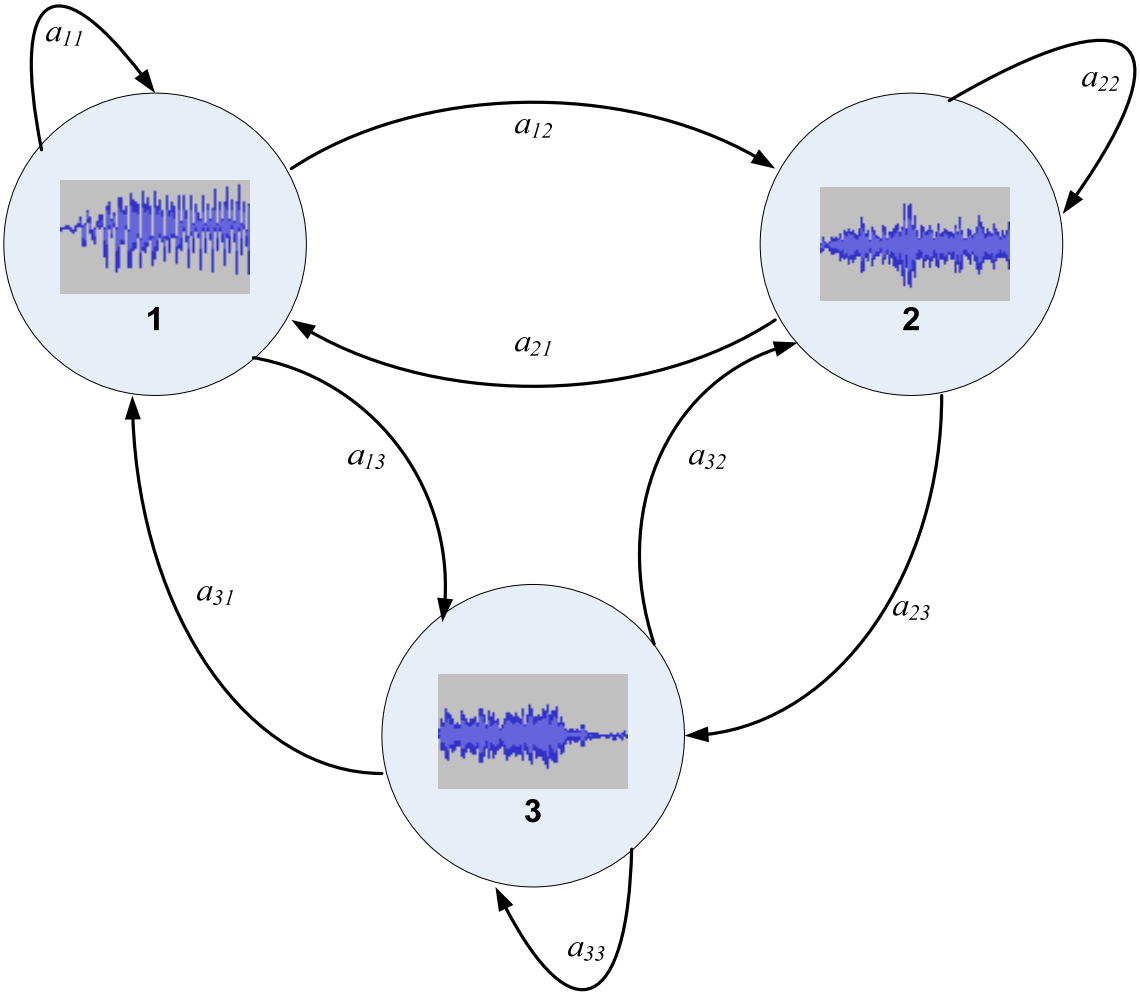
MFCC'leri elde etmek için *M* adet üçgen filtreden oluşan bir filtre kümesi oluşturulur. Şekil 2.3'de filtre kümesi genel yapısı görülmektedir. Her filtrenin log-enerji çıkışı hesaplanır ve *M* adet filtre çıkışının DCT uygulanarak MFCC'ler elde edilir. MFCC aşağıdaki şekilde hesaplanır.

$$c[n] = \sum_{m=0}^{M-1} S[m] \cos\left(\frac{\pi n\left(\frac{m-1}{2}\right)}{M}\right) \qquad 0 \le n \le M \tag{2.6}$$

2.1.2 Saklı Markov Modelleri

Saklı Markov Modeli konuşma gibi istatistiksel özellikleri zamanla değişkenlik gösteren dizilerin modellenmesinde kullanılır. Saklı Markov modelinde durumlar doğrudan gözlenemez. Gözlem dizilerine en iyileme teknikleri uygulanarak en yüksek olasılıklı durum dizisi elde edilir. Konuşma tanımada yaygın olarak kullanılması 1980'lerden sonra başlamıştır (Edizkan, 1999). Konuşma sinyali istatistiksel özellikleri zamanla değişen bir sinyaldir. Herhangi bir anlamlı ses dizisi üretmek istediğimizde gırtlak ve dil gibi ses organlarımız hava basıncını ve hava akışını duyulabilecek ses dizileri üretecek şekilde modüle ederler. Bazı sesler kHz'ler düzeyinde spektral bileşenler içerebilir. Buna rağmen ses organlarımızın yapısı saniyede en fazla 10 kere değişir. Ses modelleme, belirli seslerin kısa zaman spektral özelliklerinin analizini içerir ve bu modelleme farklı seslere karşılık gelen ses

organ yapısının uzun zaman değişimini tanımlamamızı sağlar. Zamanla değişkenlik gösteren ve spektral gözlem dizileri ile temsil edilen ses dizilerini tanımlayabilmenin bir yolu bu diziyi bir sesten diğer bir sese geçiş şeklinde Markov zincirleri ile göstermektir. Şekil 2.4'de görülen birinci derece 3 durumlu Markov zincirinde, sistemin bir *t* anında *N* farklı durumdan birinde (S_1, S_2, S_3, S_4, S_N) olacak şekilde tanımlanabilir.



Şekil 2.4 Birinci derece 3 durumlu Saklı Markov modeli

Sistemin *t* anındaki durumu *qt* durum değişkeni ile gösterelim. Markov zinciri durum geçiş olasılığı $A=\{aij\}$ matrisi ile gösterilir. Durum geçiş olasılık matrisindeki elemanlar ise aşağıdaki gibi tanımlanır.

$$a_{ij} = P(q_t = S_j \,|\, q_{t-1} = S_i) \quad 1 \le i, \quad i \le N \qquad (2.7)$$

Denklem (2.7)'de geçiş olasılıklarının zamana bağımlı olmaması için markov zinciri homojen olduğu kabul edilmiştir. Geçiş olasılıklarında tüm i'ler için $a_{ij} \geq 0$ ve $\sum_{j=1}^{N} a_{ij} = 1$ kısıtlaması vardır. Sistemin başlangıç durumu olan $t = 0$ anında $\pi' = [\pi_1, \pi_2, ..., \pi_N]$ başlangıç durumu olasılık vektörü ile gösterilir.

SMM'nin sahip olduğu tüm parametreler aşağıda listelenmiştir.

1. Durum (state) sayısı N, $S=\{S_1, S_2, ..., S_N\}$
2. Gözlem sembolleri, bunlar $V=\{V_1, V_2, ..., V_M\}$
3. Durum geçiş olasılığı, $A=\{a_{ij}\}$
4. Gözlem sembolü olasılık dağılımı, $B=\{b_j(k)\}$.
5. İlk durum olasılığı $\pi=\{\pi_i\}$.

Model parametreleri aşağıdaki şekilde ifade edilir.

$$\lambda = (A, B, \pi) \qquad (2.8)$$

Burada t=1'den t=T'ye kadar gözlem (observation) dizisi aşağıdaki gibi gösterilecektir.

$$\mathbf{O} = O_1 O_2 ... O_T \qquad (2.9)$$

2.1.3 SMM'ler için üç temel problem

Saklı Markov model'in uygulanmasında üç ana problemin çözümü ile ilgilenilir (Rabiner, 1989). Birinci problem, verilen gözlem dizisi O ile λ modeli için gözlemin model tarafından üretilme olasılığı $P(O|\lambda)$'nın etkin olarak hesaplanmasıdır. İkinci problem ise verilen gözlem dizisi O ile λ modeli için en yüksek olasılıklı q durum dizisinin bulunmasıdır. Yani modelin gizli kalmış yapısının keşfedilmesidir. Üçüncü problem ise, $P(O|\lambda)$ olasılığını en yüksek hale getirmek için model parametrelerinin tayin edilmesidir.

Bu üç problem sırasıyla değerlendirme problemi, model yapısını öğrenme ve kestirim (estimation) problemi olarak tanımlanır.

2.1.3.1 Değerlendirme problemi- İleri geri işlem (Forward-backward)

Değerlendirme problemi bir modelle gözlemlerin ne kadar uyuştuğunu gösteren bir değerdir. Eğer çeşitli modeller arasında en iyi olanı seçilmeye çalışılıyorsa, değerlendirme probleminin çözümü gözlemle uyuşan en iyi modeli verecektir. İki kısımdan oluşur. İleri işlem için, ileri değişkeni $\alpha_t(i)$ aşağıdaki gibi tanımlanır.

$$\alpha_t(i) = P(O_1 O_2 ... O_T, q_t = S_i | \lambda) \qquad (2.10)$$

Böylece, $\alpha_t(i)$, t'ye kadar olan kısmi gözlem sırasının i durumu ve t zamanında verilen λ modeli için olasılığını gösterir. Bunu $\alpha_i(t)$ için tümevarım yöntemiyle aşağıdaki gibi çözeriz.

1. Başlangıç:

$$\alpha_t(i) = \pi_i b_i(O_1), \quad 1 \le i \le N \qquad (2.11)$$

2. İşlem:

$$\alpha_{t+1}(j) = \left[\sum_{i=t}^{N} \alpha_t(i) a_{ij} \right] b_i(O_{t+1}), \quad 1 \le t \le T-1, \quad 1 \le j \le N. \qquad (2.12)$$

3. Sonuçlandırma:

$$P(O|\lambda) = \sum_{t=1}^{N} \alpha_t(i). \qquad (2.13)$$

Benzer şekilde geri işlem de tanımlanabilir. Burada geri değişken $\beta_t(i)$ dikkate alınmalıdır. Burada değişken $t+1$'den T'ye kısmi gözlemin olasılığını hesaba katmaktadır. Aşağıdaki gibi yazılabilir:

$$\beta_t(i) = P(O_{t+1}O_{t+2}...O_T, q_t = S_i, \lambda) \tag{2.14}$$

$\beta_t(i)$ için aşağıdaki gibi çözeriz.

1. Başlangıç:

$$\beta_t(i) = 1, \quad 1 \le i \le N \tag{2.15}$$

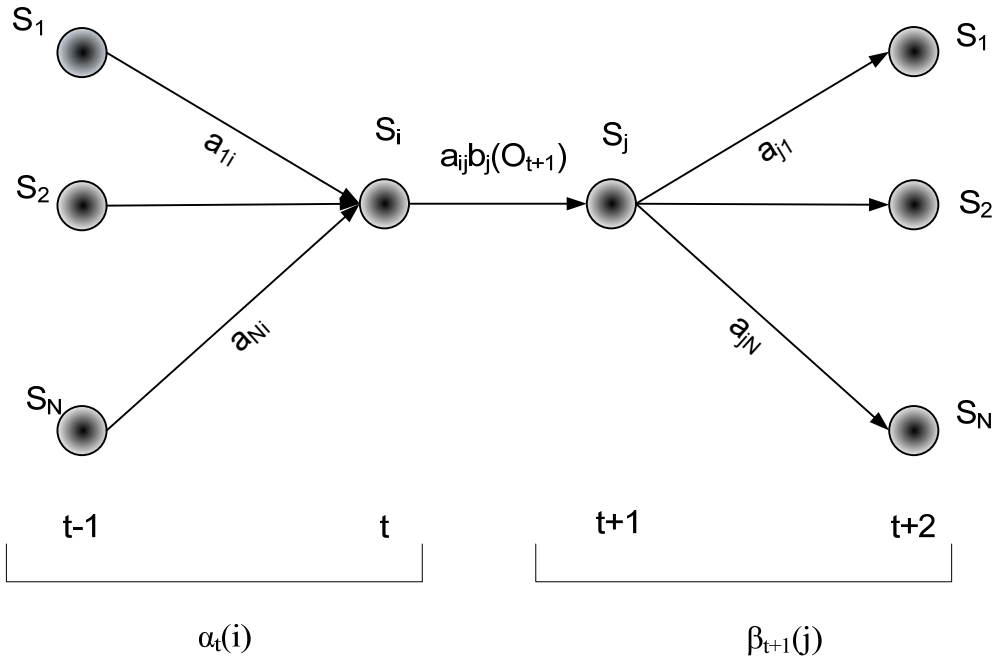
2. İşlem:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_i(O_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, ..., 1 \qquad 1 \le i \le N. \tag{2.16}$$

3. Sonuçlandırma:

$$P(O|\lambda) = \sum_{i=1}^{N} \beta_1(i). \tag{2.17}$$

Şekil 2.5'de $P(O/\lambda)$'yı elde etmek için ileri geri olasılık fonksiyonları verilmektedir.



Şekil 2.5 P(O/λ)'yı elde etmek için ileri geri olasılık fonksiyonları

2.1.3.2 Model yapısını öğrenme-Viterbi algoritması

SMM'lerde model yapısı gizlidir. Genelde O gözlem dizisini üreten en yüksek olasılıklı durum dizisinin ne olduğunun tespit edilmesi ile ilgilenilir. Saklı Markov modelde tanımlanan olasılık değeri, durum dizisinin ne olduğuna doğrudan bağlı olarak hesaplanmamasına rağmen, birçok uygulamada en yüksek olasılıklı durum dizisinin ne olduğunun bilinmesi çeşitli nedenlerden gereklidir. Örneğin, bir kelime içindeki sesleri göstermek için bir kelime modelinin durumları kullanılırsa, kelime içindeki sesler ile ses bölütleri arasındaki benzerlik bilinmek istenebilir. Belli ses bölütlerinin devam süreleri ses tanıma amaçları için faydalı bir bilgi sağlar. Ayrıca en yüksek olasılıklı durum dizilerine bakılarak model topolojisinde değişiklik yapılabilir.

Bu problemin çözümünde, mümkün olan q dizileri üzerinden, $P(q|O,\lambda)$ olasılığını en büyük hale getirmek amaçlanır. İkinci problemin çözümünde, verilen modele göre gözlem dizisini en yüksek olasılıkla veren durum dizisinin belirlenmesinde Viterbi gibi dinamik programlama metotları kullanılır (Edizkan, 1999). $P(q|O,\lambda)$ 'nın en büyük olması

$P(q,O|\lambda)$'nın en büyük olması demektir, çünkü eniyilemede $P(O|\lambda)$ ifadesi bulunmaz. Bu problemin çözümünde, mümkün olan q dizileri üzerinden, $P(q|O,\lambda)$ olasılığını en büyük hale getirmek amaçlanır. Bunun için Viterbi gibi dinamik programlama metotları kullanılmalıdır. $P(q|O,\lambda)$'nın en büyük olması $P(q,O|\lambda)$'nın en büyük olması demektir, çünkü eniyilemede $P(O|\lambda)$ ifadesi bulunmaz. Onun yerine aşağıdaki denklemden

$$\begin{aligned} &P(q_1,q_2,...,q_t,O_1,O_2,...,O_t|\lambda) \\ &= P(q_1,q_2,...,q_{t-1},O_1,O_2,...,O_{t-1}|\lambda)\cdot a_{q_{t-1}q_t} b_{q_t}(O_t) \end{aligned} \qquad (2.18)$$

olduğu görülmektedir.

İlk t gözlemini hesaba katan ve S_i durumunda sonlanan tek bir yol boyunca en yüksek olasılığı $\delta_t(i)$ olarak tanımlayalım.

$$\delta_t(i) = \max_{q_1,q_2,...,q_{t-1}} P(q_1,q_2,...,q_t = S_i, O_1,O_2,...,O_t|\lambda) \qquad (2.19)$$

Bu ifadeden yola çıkarak $\delta_{t+1}(i)$ aşağıdaki gibi ifade edilir.

$$\delta_{t+1}(i) = [\max_i \delta_t(i)a_{ij}]b_j(O_{t+1}) \qquad (2.20)$$

Böylece en iyi durum dizisi, $\delta_T(q_T)$ değeri ile biten bir durum dizisidir. Üstteki denklem Viterbi algoritması için uygun bir denklemdir.

Viterbi algoritması, en iyi durum sırasını bulur, böylece üstteki denklemin her *t* zamanı ve *j* durumu için geçerli olmasını sağlar. Bu amaçla, $\psi_1(i)$ terimi tanımlanır.

1. Başlangıç:

$$\delta_1(j) = \pi_i b_i(O_1), \quad 1 \le i \le N$$
$$\psi_1(i) = 0 \qquad (2.21)$$

2.Yineleme

$$\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i)a_{ij}]b_j(O_t) \quad 2 \le t \le T$$
$$1 \le j \le N.$$

$$\psi_t(j) = \arg\max_{1 \le i \le N}[\delta_{t-1}(i)a_{ij}] \quad 2 \le t \le T \qquad (2.22)$$
$$1 \le j \le N.$$

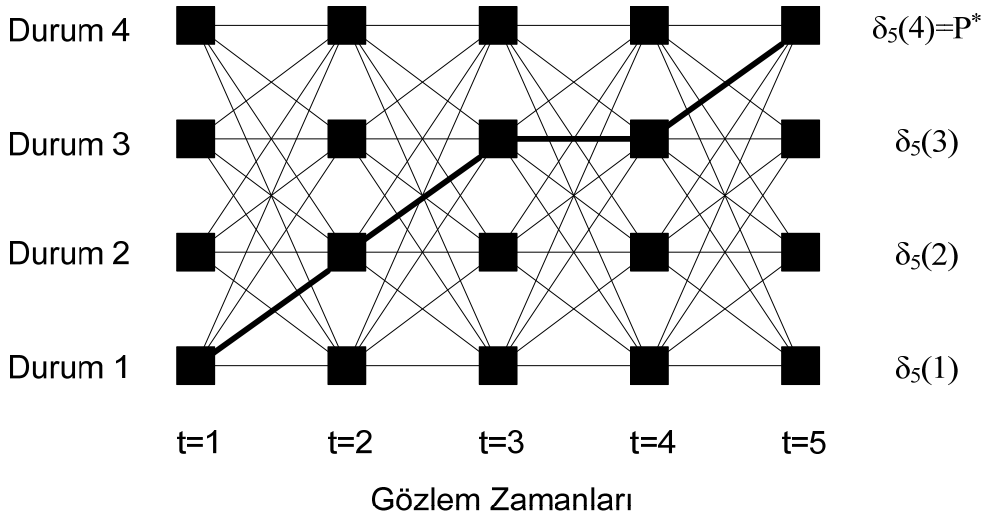
3. Sonuçlandırma:

$$P^* = \max_{1 \le i \le N}[\delta_T(i)]$$
$$q_T^* = \arg\max_{1 \le i \le N}[\delta_T(i)] \qquad (2.23)$$

4. Yol (durum sırası) tahsisi

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, ..., 1 \qquad (2.24)$$

Viterbi algoritması, ileri ileri işlemine benzer. İki arasında fark sadece toplam yerine maksimumun alınma işleminin yapılıyor olmasıdır. Viterbi ile en yüksek olasılığı veren yol üzeriden $P(O|\lambda)$ olasılığı elde edilir. Bu olasılık değeri (P^*) sınıflamada da kullanılabilir. Şekil 2.6'de N=4 durumlu bir SMM'de T=5 için en yüksek olasılıklı yol koyu olarak gösterilmektedir.



Şekil 2.6 Viterbi akış algoritması.

2.1.3.3 Kestirim- Baum-Welch metodu

Verilen bir gözlem dizisi için tahmin probleminin çözümü bu diziyi en yüksek olasılıkla üretecek doğru model parametrelerini bulmamızı sağlar. Ses tanımada bu işlem eğitim olarak adlandırılır. Model parametrelerini elde etmek için kullanılan gözlem dizisi de eğitim dizisi olarak tanımlanır. Tahmin problemi için gözlem dizisinin model tarafından üretilme olasılığını en yüksek yapan analitik bir çözüm mevcut değildir. Bununla beraber $P(O|\lambda)$ olasılığını yerel olarak en yüksek olarak elde eden Baum-Welch yöntemi veya gradient tekniği gibi özyineleme işlemleri kullanılarak $\lambda = (\pi, A, B)$ model parametreleri seçilebilir. Tahmin probleminin çözümünde en büyük olabilir yöntemi kullanılır.

Baum-Welch uygulamasında bazı olasılık değerlerinin bulunması gerekir. Bu olasılık değerleri bulunduktan sonra model parametrelerini eniyilenmiş değerleri hesaplanabilir. $\xi_t(i,j)$ olasılığı t zamanında S_i durumunda, $t+1$ zamanında ise S_j durumunda bulunma olasılığı olarak tanımlanır.

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_i | O, \lambda) \qquad (2.25)$$

İleri geri değişkenler kullanarak $\xi_t(i,j)$ terimi aşağıdaki şekilde ifade edilir:

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)}$$

$$= \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \qquad (2.26)$$

Buna ek olarak, verilen gözlem dizisi için t zamanında S_i durumunda bulunma olasılığı $\gamma_t(i)$, $\xi_t(i,j)$ ile aşağıdaki şekilde ilişkilendirilir.

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j) \qquad (2.27)$$

Zaman ekseni boyunca terimleri toplarsak, aşağıdakiler elde edilir.

S_i'den beklenen geçiş sayısı $\sum_{t=1}^{T-1}\gamma_t(i)$

Beklenen S_i'de kalması sayısı $\sum_{t=1}^{T}\gamma_t(i)$

S_i'den S_j'ye beklenen geçiş sayısı $\sum_{t=1}^{T-1}\xi_t(i,j)$

Kesikli SMM parametrelerinin (π, A ve B) eniyilenmesi aşağıdaki ifadeler kullanılarak yapılır. Sürekli yoğunluk SMM için eniyilemenin nasıl yapılacağı Rabiner'de (1989) verilmektedir.

$\bar{\pi}_i = S_i'$ de (t=1) anında bulunma sayısı $= \gamma_1(i)$

$$\bar{a}_{ij} = \frac{S_i\text{'den } S_j\text{'ye beklenen geçiş sayısı}}{S_i\text{'den beklenen geçiş sayısı}}$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} \qquad (2.28)$$

$$\overline{b}_j(k) = \frac{S_j\text{'de beklenen buluma sayısı ve gözlem sembolü } v_k}{S_j\text{'de beklenen bulunma sayısı}}$$

$$\overline{b}_j(k) = \frac{\sum_{\substack{t=1 \\ O_t = v_k}}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(i)} \qquad (2.29)$$

Yeni model parametreleri $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$ ile gösterilir. Baum-Welch metodu, verilen gözlem dizisinin model tarafından üretilme olasılığın artışı garanti eder. Eğitime yeni ve bir önceki modelden elde edilen olasılık arasındaki fark belli bir değere yakınsayıncaya kadar devam edilir.

BÖLÜM 3

HTK

3.1 Giriş

HTK, konuşma tanıma sistemleri oluşturmak için kullanılır. Sürekli veya yarı sürekli yoğunluklu, ya da kesikli olasılıklı SMM tabanlı işlemler gerçekleştirir. Cambridge University Speech Group tarafından geliştirilmiş ve 1990'ların başından beri güncellenmektedir. HTK kaynak kodları açık kaynak kodunda olup platforma göre derlenmektedir (HTK web sitesi, http://htk.eng.cam.ac.uk).

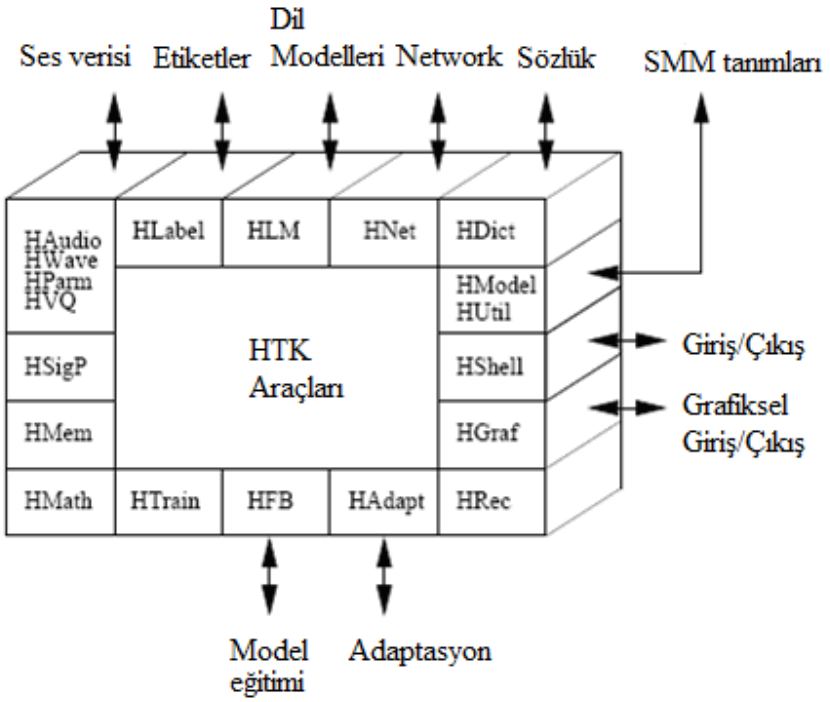
HTK, SMM sistemlerinin araştırılması ve geliştirilmesini destekleyecek şekilde esnek olarak tasarlanmıştır. Yazılım komutlar ve seçenekler ile gerekli işlemleri yapacak şekilde kontrol edilir. Yalıtık veya sürekli konuşma tanıma sistemleri kelime tabanlı veya alt-kelime tabanlı modeller kullanarak yapılır. HTK data kodlama, Baum-Welch yinelemeli metodu (reestimation), Viterbi çözümlemesi (decoding) metodlarını kullanan SMM eğitim işlemlerini yapan birçok araç içerir.

HTK aynı zamanda dil modeli oluşturmayı desteklemektedir. Dil modeli n-gram tabanlıdır. Versiyon 3.4.1 ile HDecode trigram dil modelini destekler hale gelmiştir.

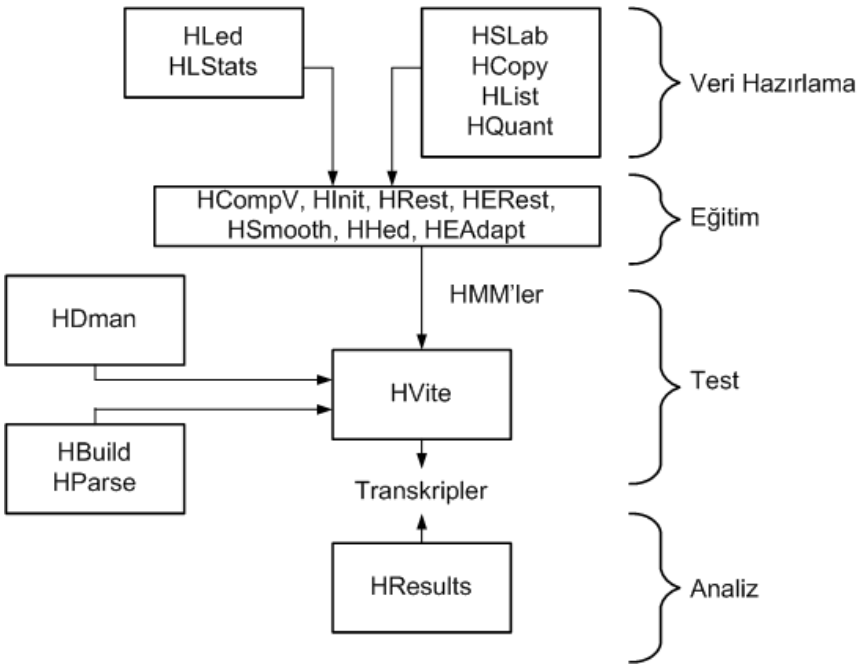
3.2 Araçlar ve Modüller

Konuşma tanıma sistemi dört aşamada gerçeklenir: veri hazırlama, eğitim, test ve analiz. HTK'yı ve yan modülleri kullanmak için bu aşamalar için hazırlanmış komutlar vardır. Komutlar kısaltılmış komut formundadırlar. Bu formlar işletim sistemi ortamında çalıştırılan komutları simgeler. Komutlar bir işlemi yaparken modülleri kullanır. Bir açıdan modüller iç içe geçmiş durumdadır. Kullanıcı komut satırında veya metin dosyasında seçenekleri kullanarak bu modülleri yönetir. Şekil 3.1 ve Şekil 3.2'de HTK'deki modüller görülmektedir.

Temel olarak HTK metin ve konuşma veri dosyası olarak iki tip dosya ile çalışır. Metin dosyaları işlem komutlarını, yapılandırma parametrelerini, konuşma dosyalarının transkripsiyonlarını veya işlem yaparken kullanılacak dosyaların listesini içerir.



Şekil 3.1 HTK yazılım mimarisi



Şekil 3.2 HTK işlem basamakları

3.2.1 Veri Hazırlama Araçları:

Konuşma tanıma sistemi geliştirme için, konuşma verisi ve ilişkili transkripsiyonlar gerekmektedir. Veritabanının Saklı Markov Modelleri (SMM'leri) eğitmek için kullanılmasından önce, uygun formata dönüştürülmesi gerekir.

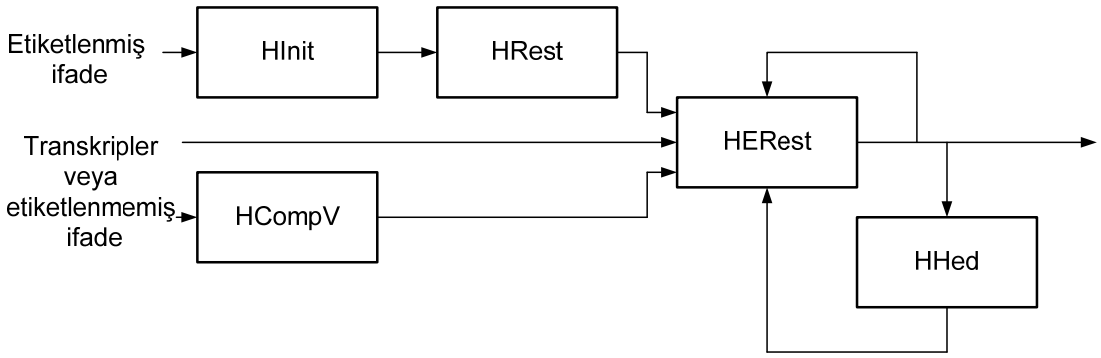
HCopy: Ses dosyalarını parametrelere dönüştürmek için kullanılır. Ses dosyası üzerinde kopyalama işlemi yapar ve kopyalarken gerekli parametrik forma dönüştürür. Tüm dosyayı kopyalama yanında, uygun yapılandırma parametreleri ile ilgili segmentlerin ve dosyaların bitiştirilmesine olanak sağlar.

HList: Konuşma dosyalarının ve parametrik çevrimin içeriğini kontrol eder.

HLed: Transkripsiyon dosyalarının HTK etiket dosyalarına dönüştürülmesi için kullanılır.

3.2.2 Eğitim Araçları:

Konuşma tanıyıcı sistemlerde bir sonraki adım, her SMM'nin ilk örnek olarak tanımlanması gerekir. HTK, SMM'lerin rastgele topolojide üretilmesine olanak sağlar. HMM prototipleri metin dosyası formatındadır ve basit bir metin editörü ile düzenlenebilir. Prototipler, SMM'leri genel karakteristikleri ve topolojisi ile gerçek parametrelerin eğitim araçları ile hesaplanması için düzenlenmiştir. Geçiş olasılıkları kabul edilebilir değerlerden seçilir. Fakat eğitim işlemi bunlardan çok fazla etkilenmez. Basit bir yaklaşım tüm geçiş olasılıklarını eşit almak olabilir. SMM'lerin eğitilmesi birçok basamaktan oluşur. HTK ile eğitim basamakları Şekil 3.3'de görülmektedir.



Şekil 3.3 HTK ile eğitim

İlk adım olarak modellerin başlangıç değerleri atanır. Fonem sınır değerleri mevcut ise bu önyükleme verisi olarak kullanılabilir. Bu durumda **HInit** ve **HRest** araçları kullanılır. SMM'ler ayrı ayrı oluşturulur. HInit, önyükleme verisini okur ve istenen fonemleri ses verisinden bulur ve yinelemeli olarak bir parametre kümesini hesaplar. İlk yinelemede eğitim verisi eşit dağılımlı (flat start) olarak bölünür, ilgili veri bölümü ile her model durumu eşlenir. Bu eşleşme sonucu durumlardaki gözlemlerin ortalaması ve varyansı hesaplanır. Daha sonraki yinelemelerde eşit dağılım yerine Viterbi hizalaması kullanılır. **HInit** ilk değerleri hesapladıktan sonra, **HRest** ile değerler tekrar hesaplanır. **HRest** de önyükleme verisini kullanır. Eğitimde yukarıda bahsedilen Baum-Welch yinelemeli metodunu kullanır. Eğer önyükleme verisi mevcut değil ise, düz bir başlangıç için **HCompV** kullanarak yapılabilir. Bu durumda, tüm SMM'ler genel bir ortalama ve varyans değeri ile ortak başlangıç ile eğitime girer. İlk değerler atandıktan sonra, tüm eğitim seti kullanılarak **HERest** aracı ile gömülü bir eğitim yapılır. **HERest**, tüm SMM'ler için aynı ayda bir tek Baum-Welch hesaplaması ile yeni değerleri hesaplar. Her eğitim ifadesi için, ilgili fonem modelleri birbirine eklenir ve daha sonra ileri-geri algoritma ile sıralamadaki tüm SMM'ler için durum işgali istatistikleri, ortalamaları, varyansları vs., hesaplanır. Tüm eğitim verisi işlendikten sonra, toplanan istatistikler SMM parametrelerinin yeni değerlerinin atanması için kullanılır.

HERest, ana HTK aracıdır ve değişik seçenekler ile kullanılabilir. Genel anlamda, ilk olarak metinden bağımsız modeller üretilir. Daha sonra yinelemeli olarak metin-bağımlı ikili ve üçlü fonemler (bifon, trifon vs.) fonemler sisteme dahil edilir.

3.3 Dosya Tipleri

Konuşma verisi içeren dosyalar dışında HTK tamamiyle metin dosyaları ile çalışır. Fonksiyonlarına göre değişik uzantılara sahip olsalar bile herhangi bir editör programı ile değiştirilebilirler.

Konuşma dosyaları A-law/μ-law 8-bit (CCITT standard) "wav" formatında olmalıdır.

HSLab aracı kullanıcının konuşmayı kaydetmesine ve etiketlemesine olanak sağlar. Etiketlemek konuşmaya bir transkript atamak demektir.

3.3.1 Etiket dosyaları:

Bir etiket dosyası bir konuşmanın içeriğini tutar. Konuşmanın özellik vektörleri bir kelime veya fonem ile ilişkilendirilmesi ve SMM'in oluşturulması için etiketlemek mutlaka gereklidir. Etiket dosyaları eğitim esnasında veya tanıma sonuçlarında kullanılabilir. HTK'de tanınacak cümle ve tanıma sonucu elde edilen cümle kıyaslanır ve tanıma oranları hesaplanır.

Etiketleme dört seviyede olabilir: Kelime, fonem, üçlü-fonem veya ikili-fonem, ve hece. Etiket dosyasında etikete ait konuşma bölümü tanımlanabilir. Etiket dosyasındaki her satır etiketlenmiş bölümün başlangıç ve bitiş zamanlarını içerir.

3.3.2 Sözlük dosyası:

Sözlük dosyasının her satırda bir kelime ve onun okunuşu yer alır. Bazen aynı kelimenin farklı okunuşları olacağından, her farklı okunuş da dosyada listelenir.

sil *[] SIL*
SENT-START *[] SIL*
SENT-END[] SIL
bir *B IY RH*
iki *IY KK IY*
iki *IY K IY*
UC *UE CH*

dOrt D OE R T

beS B E SH

…

3.3.3 SMM tanım dosyası:

HTK'de akustik modele karşılık gelen SMM tanım dosyaları vardır. HTK'da tanım dosyalarının belli bir formatı vardır. Aşağıda A fonemine ait SMM tanım dosyasının baş kısmını görülmektedir.

~o <VecSize> 39 <MFCC_E_D_A_Z>

~h "A"

<BeginHMM>

<NumStates> 5

<State> 2

<Mean> 39

…

3.3.4 Konfigürasyon dosyası:

Kullanıcı tarafından belirlenen parametreleri içerir. Burada SOURCEKIND ve SOURCE FORMAT ile kayıtların WAV dosyası olduğunu belirtilir. MFCC_0_D_A ise MFCC'leri oluştururken MFCC'nin sıfırıncı katsayısının,delta ve acceleration değerlerinin kullanılacağını gösterir. FFT, hamming pencerelenmiş ve 0.97 değeri ile önvurgulanmış veriler üzerine uygulanır. 25 milisaniye uzunluğunda pencerede 10 milisaniyelik (HTK 100 nanosaniyelik birimleri kullanır) kayma kullanılacağı belirtilmektedir. Filtre sayısı 26'dır ve 12 MFCC katsayısı çıkarılacaktır. ENORMALISE değişkeni kayıtlı ses dosyalarında enerji düzenlenmesi sağlar.

SOURCEKIND=WAVEFORM

SOURCEFORMAT=WAV

TARGETKIND = MFCC_0_D_A

TARGETRATE = 100000.0

SAVECOMPRESSED = T

SAVEWITHCRC = T

WINDOWSIZE = 250000.0

USEHAMMING = T

PREEMCOEF = 0.97

NUMCHANS = 26

CEPLIFTER = 22

NUMCEPS = 12

ENORMALISE = T

USEPOWER=T

3.3.5 Skript dosyası:

Birden fazla dosya ismi isteyen araçlar için, içinde dosya isimlerini listeleyen metin dosyalarıdır.

Konuşma verilerinin MFCC'lerini çıkarmak için hazırlanmış bir skript dosyası aşağıdaki gibidir:

data/train/wav/s1075-000.wav data/train/mfcc/s1075-000.mfcc

data/train/wav/s1075-001.wav data/train/mfcc/s1075-001.mfcc

data/train/wav/s1075-002.wav data/train/mfcc/s1075-002.mfcc

...

3.3.6 Toplu işlem dosyası:

Tekrar tekrar kullanılan araçlar için (**HInit** veya **HRest** gibi) toplu işlem dosyaları oluşturulur içerisinde aşağıdakine benzer uzun komutlar sırasıyla yer alır.

$HOME/bin/HRest -A -D -T 1 -S training/trainlist.txt -M model/HMM1 -H model/SMM0/HMM_A.txt -L data/train/lab A

$HOME/bin/HRest -A -D -T 1 -S training/trainlist.txt -M model/HMM1 -H model/SMM0/HMM_AA.txt -L data/train/lab AA

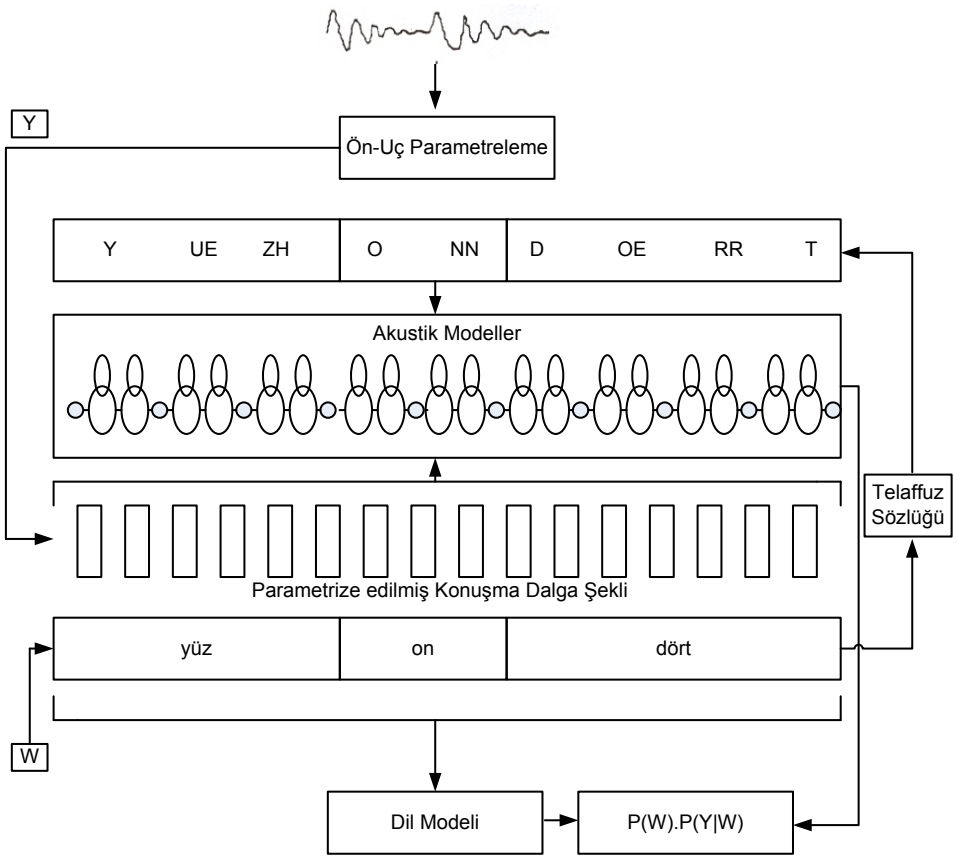
$HOME/bin/HRest -A -D -T 1 -S training/trainlist.txt -M model/HMM1 -H model/SMM0/HMM_B.txt -L data/train/lab B

3.3.7 HTK ile İstatistiksel Konuşma Tanıma

Bu bölümde, Cambridge Üniversitesi Hidden Markov Tool Kit (HTK) (Young et al., 2009) sisteminde konuşma tanıma sisteminin nasıl olduğu anlatılmaktadır.

Bilinmeyen konuşma sinyali ön-uç sinyal işlemcisi tarafından $Y=y_1, y_2, \ldots , y_T$ gibi akustik vektörlere dönüştürülür. Bu vektörlerin her biri 10 msn'lik kısa-süreyi kapsayan konuşma spektrumunu simgeler. Böylece 3 saniye süren 10 kelimelik ifade yaklaşık 3 saniye sürmekte ve T=300 akustik vektör ile gösterilmektedir.

Şekil 3.4 bu ilişkilerin nasıl hesaplandığını göstermektedir. Şekilde kelime dizisi *W*="Yüz on dört" için dil modeli *P(W)* olasılığını hesaplar. Daha sonra, kelime dizisindeki her kelime telaffuz sözlüğü kullanılarak temel ses veya fonem sırasına dönüştürülür. Her temel ses veya fonem Saklı Markov Model (SMM) ile istatistiksel olarak modellenir.



Şekil 3.4 İstatistiksel konuşma tanıma

Örnek ifadeyi oluşturmak için SMM dizileri birleştirilerek tek bir bileşik oluşturulur, daha sonra gözlenen *Y* dizisi için modelin olasılığı hesaplanır. Bu olasılık *P(Y|W)* değerini verir. Teoride bu işlem her muhtemel kelime sırası için tekrarlanır.

Yukarıda tanımlanan tasarım felsefesini pratik bir sisteme dönüştürmek, bir miktar zorlu problemin çözümüne ihtiyaç duyar.

İlk olarak, konuşma sinyalinden SMM–tabanlı akustik modelleri ile uyumlu gerekli akustik bilgiyi oluşturacak ön-uç parametrelemeye ihtiyaç vardır.

İkinci olarak, SMM modellerinin her sesin dağılımını, sesin geçtiği her cümle parçası için doğru olarak temsil etmesi zorunludur. Ayrıca, SMM parametreleri veriden tahmin edilmelidir. Ancak tüm muhtemel cümle parçalarını kapsayacak yeterli veri elde etmek mümkün değildir.

Üçüncü olarak, doğru kelime tahmini oluşturacak dil modeli tasarlanmalıdır. Yine de SMM'ler için, veri seyrekliği her zaman mevcut bir problemdir. Dil modeli eğitim verisinde yer almayan kelime dizileri ile uğraşmak zorunda kalır.

Son olarak, $\hat{W}$'yi elde etmek yukarıda listelenen iş adımlarının, tüm muhtemel kelime sıraları için uygulanması pratik değildir. Bunu yerine, potansiyel kelime dizileri aynı anda incelenerek, muhtemel olmayanları atarak işlem yapılmaktadır. Bu işleme "kodçözme (decoder)" adı verilmekte olup verimli bir decoder'in tasarımının pratik konuşma tanıma sisteminin gerçekleştirilmesi açısından önemi büyüktür.

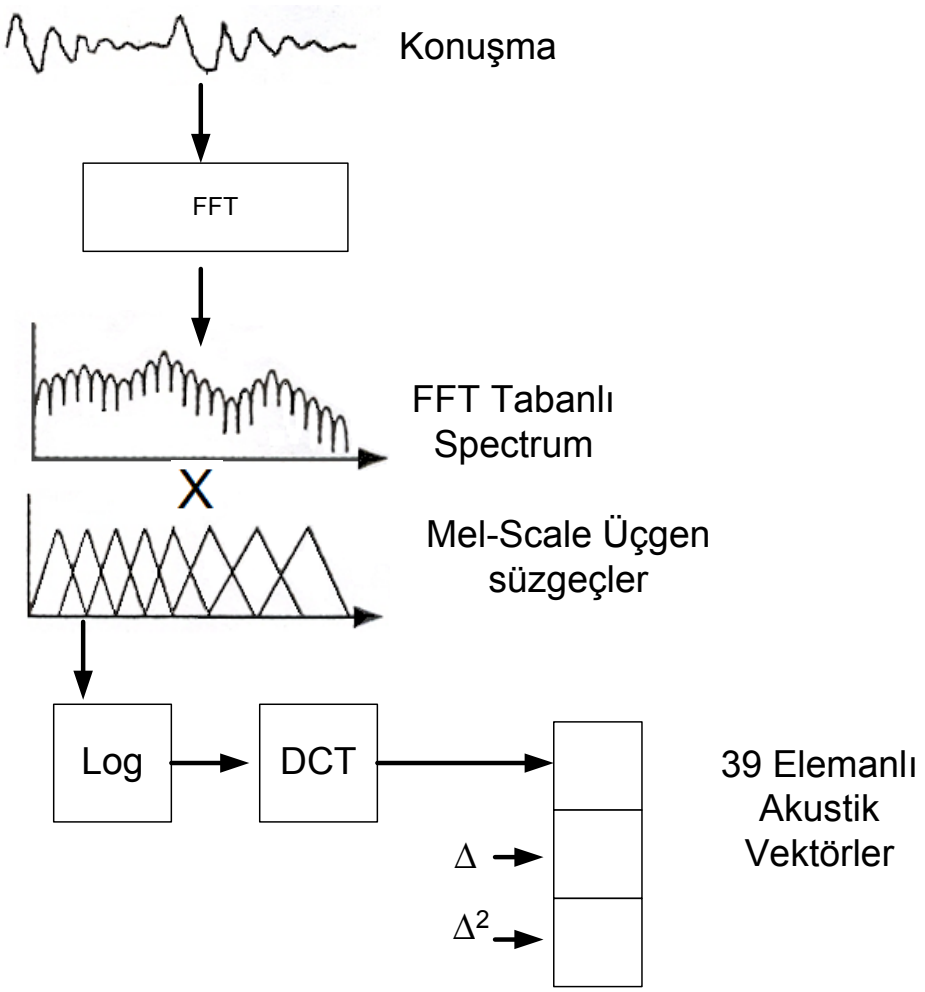
Bundan sonraki bölümlerde bu dört kısım daha ayrıntılı olarak açıklanmaktadır.

3.3.7.1 HTK'da ön-uç parametreleme

HTK'da yapılan işlemlerin ilki olan ön-uç parametreleme adımının ilk işlevi konuşma girdisini bloklara bölme ve her bloktan düzgün bir spektral kestirim oluşturmaktır. Bloklar 25 msn'lik uzunlukta ve aralarında 10 msn örtüşme olacak şekilde seçilir.

Bu tip işlemlerde kullanılan uçlara doğru incelen pencereler (Örnek: Hamming) her bloğa uygulanır. Ayrıca insan dudağından ışınım nedeniyle oluşan zayıflamanın etkisini gidermek için genelde konuşma sinyaline frekans yükseltmesi (preamphasis) uygulanır.

Spektral tahminler doğrusal kestirim veya Fourier analiz ile yapılır. Akustik vektörleri oluşturmak için birçok ilave dönüşümler mevcuttur. Örnek olarak Şekil 3.5'de MFCC'leri oluşturan HTK tanıma ön-uç birimi görülmektedir.



Şekil 3.5 MFCC tabanlı ön-uç işlemci (Young, 1996)

MFCC katsayılarını hesaplamak için, üçgen frekans kutularının doğrusal olmayan ve mel-scale adı verilen ölçüde düzenlenmesi neticesinde spektral katsayıların integralinin alınması ile Fourier spektrum düzeltilir.

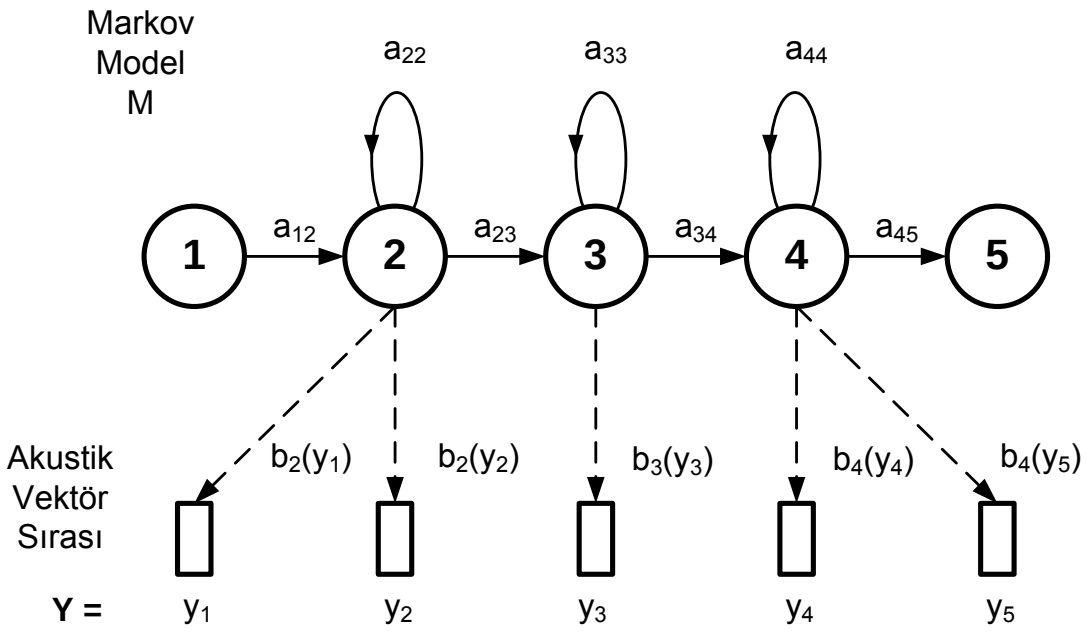
HTK tanıyıcı 8 kHz'lik konuşma için bu üçgen kutulardan 24 tane kullanır. Tahmin edilmesi istenen konuşma güç spektrumunun istatistiğini Gaussian'a yakın hale getirmek için süzgeç çıktısına log-sıkıştırma uygulanır. Son olarak da süzgeç çıktılarına Ayrık Kosinüs Dönüşümü -DCT (Discrete Cosine Transform) uygulanır. DCT bize spektral bilginin düşük mertebeli katsayılara sıkıştırılmasını sağlar. Ayrıca köşegen ortak değişinti matrisleri (diagonal covariance matrices) kullanılmasını sağlar.

HTK tanıyıcı içinde, sinyal enerjisine ilave ilk 12 cepstral katsayılar, temel 13-elemanlı akustik vektörü oluşturur. Cepstral katsayılar benzer ilişki etkisi elde edilen LP(linear prediction) katsayılardan da türetilebilir. LP katsayıları ile iyi sonuçlar da alınmıştır. Bundan sonraki bölümde de görüleceği gibi, akustik modelleme, her vektörün komşuları ile ilintisiz yani bağımsız olduğunu kabul eder. Bu zayıf bir kabuldür, çünkü insan ses üretme organları takip eden spektral kestirimler arasında devamlılık olmasını sağlar. Yine de temel statik katsayılarından elde edilen birinci ve ikinci türevler bu problemi büyük oranda azaltır. Onüç elemanlı akustik vektöre birinci ve ikinci türevler eklenerek akustik vektör eleman sayısı 39'a çıkar. Bu şekilde her pencere 39 elemanlı özellik vektörü ile temsil edilir.

3.3.8 Akustik modelleme

Akustik modellemenin amacı herhangi bir vektör *Y*'nin verilen *W* kelimesi ile olasılığını hesaplama metodunu sağlamaktır. Teoride gerekli olasılık dağılımının tespiti, her W için birçok örnek bulma ve bunlara denk gelen vektör sıralarını bir araya getirme ile yapılmaktadır. Ancak geniş-kelime hazineli sistemlerde bu pratik değildir, bunun yerine kelimeler fonem adı verilen temel seslere ayrılır. Her bir fonem bir SMM ile temsil edilir. SMM oklar ile birbirine bağlanan belli sayıda durumdan oluşmaktadır. SMM fonem modeli genelde başlangıç ve bitiş durumları ile birlikte 5 durumdan oluşmakta ve basit bir soldan sağa gösterim Şekil 3.6'de görülmektedir.

Giriş ve çıkış durumları ise modelleri birbirine bağlamayı kolaylaştırmayı sağlar. Bir fonemin çıkış durumu, diğer fonemin giriş durumu ile birleşerek bileşik SMM oluşturur. Modellerin birleşimi ile kelimeleri, kelimelerin birleşimi ise tam ifadeleri oluşturur. SMM en kolay vektör dizisi oluşturucu olarak anlaşılır. SMM bir sınırlı-durum makinesidir. Her t zaman biriminde j durumuna geçer, y_t akustik vektörü $b_j(y_t)$ olasılık yoğunluğu ile beraber oluşturulur. Ayrıca i durumundan j durumuna geçiş de olasılıklıdır ve kesikli olasılık a_{ij} ile sağlanır. Şekil 3.6 bu işlemin bir örneğini göstermektedir. Burada model X=1, 2, 2, 3, 4, 4, 5 durumlarında y_1'den y_5'e dizileri oluşturmak için hareket etmektedir.



Şekil 3.6 SMM-tabanlı fonem modeli

Vektör dizisi *Y*'nin (Young 1996) ve durum dizisi *X*'in birleşik olasılığı, *M* modeli verildiğinde, geçiş olasılıklarının ve çıkış olasılıklarının çarpımı olarak ifade edilir.

Şekil 3.6'daki *X* için:

$$P(Y, X \mid M) = a_{12} b_2(o_1) a_{22} b_2(o_1) a_{23} b_2(o_1) .... \qquad (3.1)$$

Yani *Y* ve *X=x(1),x(2),x(3),...,x(T) için*

$$P(Y, X \mid M) = a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(y_t) a_{x(t)x(t+1)} \qquad (3.2)$$

Burada *x(0)* model giriş, x(T+1) ise çıkış durumudur.

Uygulamada sadece gözlem sırası *Y* bilinir ve *X* gizlidir. Bu yüzden saklı Markov modeli olarak adlandırılır. *P(Y|M)* olasılığı tüm muhtemel durum sıraları için

denklem 3.2'den elde edilen olasılıklar toplanarak kolayca bulunur. İleri-geri (Forward-Backward) algoritma, bu iş için etkili özyinelemeli bir yöntemdir. Bu model belli bir zamanda belli bir durumda olmaya da olanak sağlar. Bu bizi Baum-Welch algoritmasına götürür. SMM, *a* ve *b* parametre setlerinin maksimum olasılıklarını basit ve etkili bir şekilde hesaplar.

P(Y|M) olasılık hesabı için Viterbi algoritması da kullanılabilir. Bu çözümlemede çok önemlidir.

SMM tabanlı fonem modelleri ile akustik ayrışım zengin kelime hazineli-konuşmacıdan bağımsız sistemler için çok önemlidir.

Denklem 3.2 logaritmik olarak *a* ve *b* terimleri ayrılarak şu şekilde yazılır.

$$\log P(Y, X \mid M) = \sum_{t=0}^{T} \log a_{x(t)x(t+1)} + \sum_{t=0}^{T} \log b_{x(t)}(y_t) \qquad (3.3)$$

Geçiş olasılıkları olan $a_{x(t)x(t+1)}$ verinin geçici yapısını modeller. Denklem (3.3)'deki her log olasılığını skor olarak alırsak, her geçiş terimi bir durumdan diğer duruma geçişin maliyeti olarak görülebilir. Bu gerçekte gerçek konuşmanın süresi için zayıf bir model sağlar. Ama bu çok önemli değildir, çünkü pratikte üstteki tanım $b_{x(t)y(t)}$ çıkış olasılığı tarafından domine edilmiştir. Her SMM durumu bir prototip akustik vektör sağlar ve log çıkış olasılığı fonksiyonu gerçek akustik vektörlerin prototip ile kıyaslanmasına olanak sağlayan bir uzaklık ölçütü verir.

İlk SMM sistemleri bir Vektörel Nicemleme (Vector Quantizer-VQ) ile birlikte kesikli bir çıkış olasılık fonksiyonu kullanmıştır. Her giren akustik vektör daha önceden hesaplanan kod kitabındaki en yakın vektörün indeks numarası ile yer değiştirmekte ve çıkış olasılık fonksiyonları sadece muhtemel VQ indexleri içeren tablolardan (look–up table) oluşmaktadır. Hesaplama olarak bu yaklaşım çok verimlidir, fakat niceleme gürültüye neden olur ve elde edilen hassasiyeti sınırlar. Bu nedenle, modern sistemler akustik vektörleri doğrudan modelleyen parametrik sürekli-yoğunluklu çıkış dağılımları kullanmaktadır. En çok kullanılan dağılım çok değişkenli karışım Gaussian denklem 3.4'de verilmektedir.

$$b_j(\boldsymbol{y}_t) = \sum_{t=0}^{T} c_{jm} N(\boldsymbol{y}_t; \mu_{jm}, \Sigma_{jm}) \qquad (3.4)$$

Burada c_{jm} karışım elemanı m'nin j durumunda ağırlığını , $N(y;\mu,\Sigma)$ ise μ ortalamaya ve Σ kovaryansa sahip çoklu değişken Gaussian'ı göstermektedir.

Buraya kadar her fonem için tek bir SMM gerektiği varsayımıyla işlem yapıldı. Türkçe için 39 (Salor Ö. vd. 2002) fonem ihtiyacı olduğu için sadece 39 SMM'nin eğitilmesi gerektiği ortaya çıkmıştır. Bazı uygulamalarda fonem sayısı 29 olarak da alınmıştır(Arslan L.M., 1999). Fakat uygulamada, metine bağlı etkiler değişik seslerin üretiminde büyük değişkenlikler oluşturur. Bu nedenle iyi bir fonetik ayrım elde etmek için her farklı kelime içi konum için farklı SMM'lerin eğitilmesi gerekir. En basit ve yaygın yaklaşım *üçlü fonem*'leri (trifon) kullanmaktır. Burada her fonemin ayrı sağ ve sol komşularına göre ayrı ayrı bir SMM modeli vardır. Mesela, *x-y+z* gösterimi *x*'den sonra ve *z*'den önce yer alan fonem *y*'yi göstermektedir. "Ara" ifadesi *SIL A R A SIL* fonem sırası ile gösterilmektedir, ve eğer üçlü fonem SMM'ler kullanılmış olsa idi *SIL SIL-A+R A-R+A R-A+SIL SIL* olarak modellenecekti.

Yukarıdaki örnekte üçlü fonem bağlamaları kelime ve cümle sınırlarını gösteren *SIL* yani "silence-sessizlik" fonemini de kapsamıştır ve *A* foneminin iki durumu da farklı SMM'ler tarafından temsil edilmiştir. Bu çapraz kelime üçlü fonemler en hassas modellemeyi sağlar, fakat daha sonra bahsedileceği şekilde kod-çözüçü'de karmaşıklığa yol açar. Daha basit sistemler kelime içi üçlü fonemlerden oluşturulur ve yukarıdaki örnek, *SIL A+R A-R+A R-A SIL* olarak ikili ve üçlü fonemlerden oluşacak şekilde modellenir.

Gaussian karışım çıkış dağılımlarını kullanmak her durum dağılımlarının çok hassas olarak modellemeye olanak sağlar. Yine de, üçlü fonemler kullanılırsa sistemin eğitmesi gereken çok fazla parametre ortaya çıkar. Örneğin geniş kelime hazineli çapraz kelime üçlü fonem sisteminin 60000 üçlü foneme ihtiyacı vardır (39 fonemle, 39^3=59319 muhtemel üçlü fonem vardır, ama dilin fonetik yapısı sonucu bir kısmı hiç görülmez.) Türkçe'deki mevcut üçlü fonemlerin tespitine yönelik yapılan bir çalışmada, günlük gazetelerden oluşturulan bir veri tabanında üçlü fonem sayısı 29266

olarak bulunmuştur (Salor Ö. vd. 2002). METUbet adı verilen fonem dizisine göre en yoğun görülen üçlü fonemler listelenmiştir.

Çizelge 3.1 Türkçe'de en sık karşılaşılan üçlü fonemler (Salor Ö.vd. 2002)

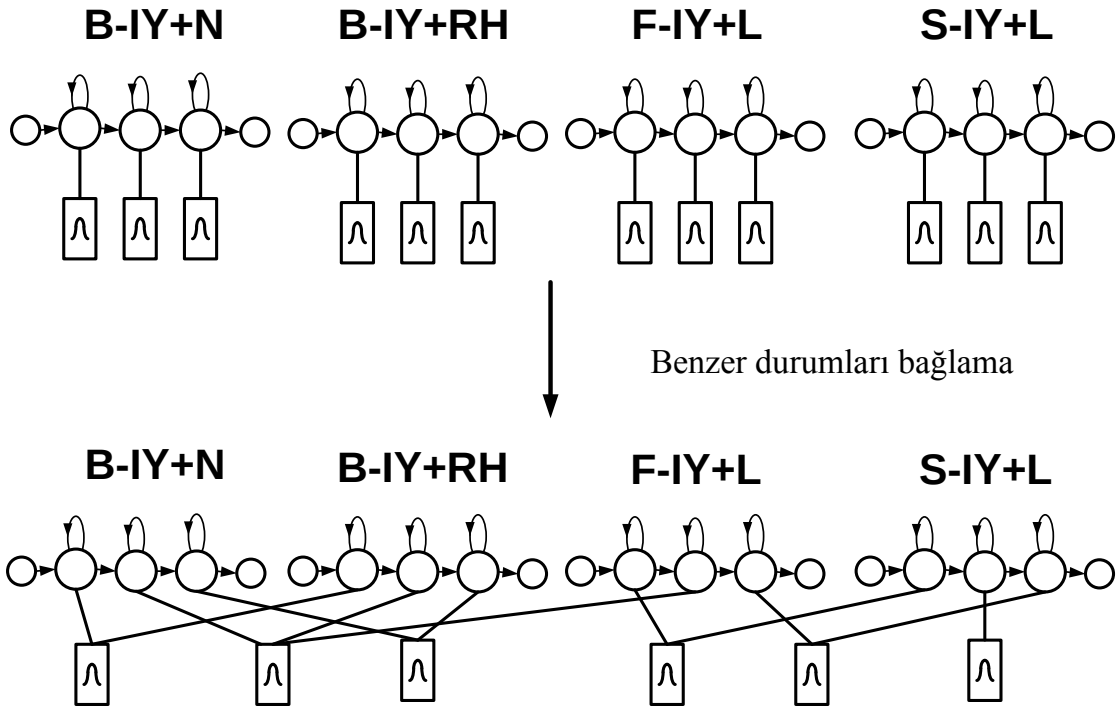
METUbet Üçlü Fonem	Örnek Kelime	Rastlama sıklığı (%)
EE RR IY	Evl***eri***ni	2.74
LL A RR	Atl***arı***nı	2.67
L EE RR	Ev***ler***i	2.61
B IY RH	**Bir**	2.54
N D AA	*yanı***nda**	2.20
IY NN IY	*Ev***ini**	1.96

Uygulamada 10 karışım elemanına sahip Gaussian dağılımlar konuşma tanıma sisteminde genelde iyi sonuç verir. Kovaryansların diagonal olduğu varsayılırsa, 39 elemanlı akustik vektörleriyle HTK tanıyıcı, her durum için 790 parametreye ihtiyaç duyar. Böylece 60000 adet üçlü fonemler toplam 142 milyon parametreye sahip olur.

Carnegie Melon Üniversitesinin geliştirdiği SPHINX sisteminde de (Lee et al. 1990) üçlü fonemler kullanmıştır.

Çok fazla parametre ve çok az eğitim verisi, istatistiksel konuşma tanıma sistemlerinin en büyük problemidir. Eski sistemler problemi tüm Gaussian elemanları bağlayarak tüm SMM durumlarının kullanacağı ortak bir havuz oluşturarak çözmekle uğraşmışlardır. Bu bağlı-karışım sistemlerde, sadece karışım elemanları ağırlıkları durumlara özeldir ve genel durumdan bağımsız olanlar ile interpolasyon yoluyla daha düzgün hale getirilebilir. Kesikli, bağlı-karışım ve sürekli-yoğunluklu SMM'ler kıyaslandığında bağlı-karışım sistemler diğerlerinden üstündür. Bunun nedeni sürekli-yoğunluklu sistemler için iyi düzleştirme tekniklerinin bulunmamasıdır. Daha yakın zamanda, parametre bağlama ile düzgünleştirme daha çok kullanılmaya başlandı. Yani durum-bağlama ve fonem tabanlı eleman bağlama üzerinde çalışıldı.

HTK tanıyıcı da durum bağlama kullanır. Buradaki mantık, akustik olarak ayrılamaz olan durumları birbirine bağlamaktır. Bu her bir durumla ilgili verinin toplanmasına olanak sağlamakta ve böylece bağlanmış durum hakkında daha sağlam tahminler vermektedir. Bu Şekil 3.7'de bu durum gösterilmektedir. Şeklin üst tarafında her üçlü fonem kendine has çıktı dağılımına sahiptir. Bağlama sonrası birçok durum dağılımları paylaşmaktadır.



Şekil 3.7 Durum bağlama

HTK tanıyıcıda, hangi durumların bağlanmasının seçimi fonetik karar ağaçları ile yapılır. Bu her fonem ve durum için "İkili ağaç" oluşturmaktır. Her ağaç'ta her düğümde "Sol bağlam genizsel mi?" gibi evet/hayır fonetik sorusu vardır. İlk olarak verilen fonem durum pozisyonu için tüm durumlar ağacın kök düğümüne konumlanır. Her cevaba bağlı olarak yeni dallara ayrılır ve bu şekilde durumlar yaprak düğümlere devam eder. Her yaprak düğümümdeki durumlar daha sonra bağlanır. Her düğümdeki

soru, durum bağlamanın en son kümesi verildiğinde eğitim verilerinin olasılığını maksimize edecek şekilde seçilmektedir.

3.4 Dil Modelleme

Dil modelinin amacı w_k kelimesinin $W_1^{k-1}=w_1...w_{k-1}$ kelime dizisi içindeki olasılığını hesaplama yöntemini oluşturmaktır. Bunun basit fakat etkili yolu N-gram'ları kullanmaktır. Burada w_k kendisinden önce gelen $n-1$ kelimeye bağlıdır.

$$P(w_k|W_1^{k-1}) = P(w_k|W_{k-n+1}^{k-1}) \quad (3.5)$$

N-gram'lar aynı anda sözdizim, anlambilim ve pragmatik'i çözer, yani anlamlı, kurallı ve mantıklı bir cümlede hangi kelimeden sonra hangi kelimenin gelme olasılığını gösterir. Bunlar İngilizce gibi anlamın kelime dizilimine bağlı olduğu dillerde çok etkilidir.

Ayrıca N-gram olasılık dağılımı, doğrudan metinden çıkarılabilir. İlave dil kurallarına ihtiyaç yoktur. Teoride, N-gram'lar basit frekans sayılarından oluşturulabilir ve bir tabloda (look-up table) saklanabilir. Örneğin trigram (N=3) için

$$P(w_k|w_{k-1}, w_{k-2}) = \frac{t(w_{k-2}, w_{k-1}, w_k)}{b(w_{k-2}, w_{k-1})} \quad (3.6)$$

$t(a,b,c)$ trigram a,b,c'nin eğitim verisi içindeki sayısı ve $b(a,b)$ bigram a,b'nin sayısıdır. Buradaki sorun V kelime içeren bir dilde V^3 muhtemel trigram bulunmasıdır. Yaklaşık 10000 kelimeli bir sistemde bile, bu çok büyük bir sayıdır. Bundan dolayı birçok trigram eğitim verisi içinde yeterince yer almaz. Denklem (3.6) çok düşük çıkar. Genelde karşılaşılan veri azlığı durumudur..

Veri azlığı durumuna çözüm discounting ve backing-off metotlarının bileşimidir. Discounting'de çok rastlanan trigramlar azaltılır ve geriye kalan olasılık dağılımı çok az görülenler arasında dağıtılır. Backing-off ise bir değer atamak için çok

az olan trigram'lar için (bir veya iki kere) kullanılır. Trigram olasılığının ölçeklenmiş bigram olasılığı ile yer değiştirmesini içerir.

$$P(w_k|w_{k-1}, w_{k-2}) = B(w_{k-1}, w_{k-2})P(w_k|w_{k-1}) \quad (3.7)$$

Trigram olasılıklarının düzgün olarak çıkarılması çok fazla dikkat gerektirmekte olduğu halde, sorunlar çözülebilir ve iyi performans alınabilir.

Türkçe için bu konuda bazı çalışmalar yapılmış ve veri tabanlarından mono-, bi- ve trigram yüzdeleri çıkarılmıştır (Dalkılıç vd, 2004). Turkish Corpus (TurCo) adı verilen veri tabanında yapılan çalışmalar neticesinde aşağıdaki tablodaki değerlere ulaşılmıştır.

Çizelge 3.2 TurCo veritabanındaki ilk sekiz mono-, bi- ve trigramlar.(Dalkılıç vd.,2004)

Rank	Mono	%	bi	%
1	ve	2.27	kültür sanat	0.13
2	bir	1.60	hava durumu	0.12
3	bu	1.29	ana sayfa	0.10
4	da	0.72	bilim teknoloji	0.10
5	de	0.71	yeni pencerede	0.09
6	için	0.52	pencerede aç	0.09
7	ile	0.40	türkiye büyük	0.08
8	türkiye	0.36	büyük millet	0.08

Rank	tri	%
1	yeni pencerede aç	0.09
2	arsiv bize ulasin	0.08
3	türkiye büyük millet	0.08
4	yazarlar kültür sanat	0.07
5	politika dünya ekonomi	0.06
6	gündem politika dünya	0.06
7	durumu astronet televizyon	0.06
8	hava durumu astronet	0.06

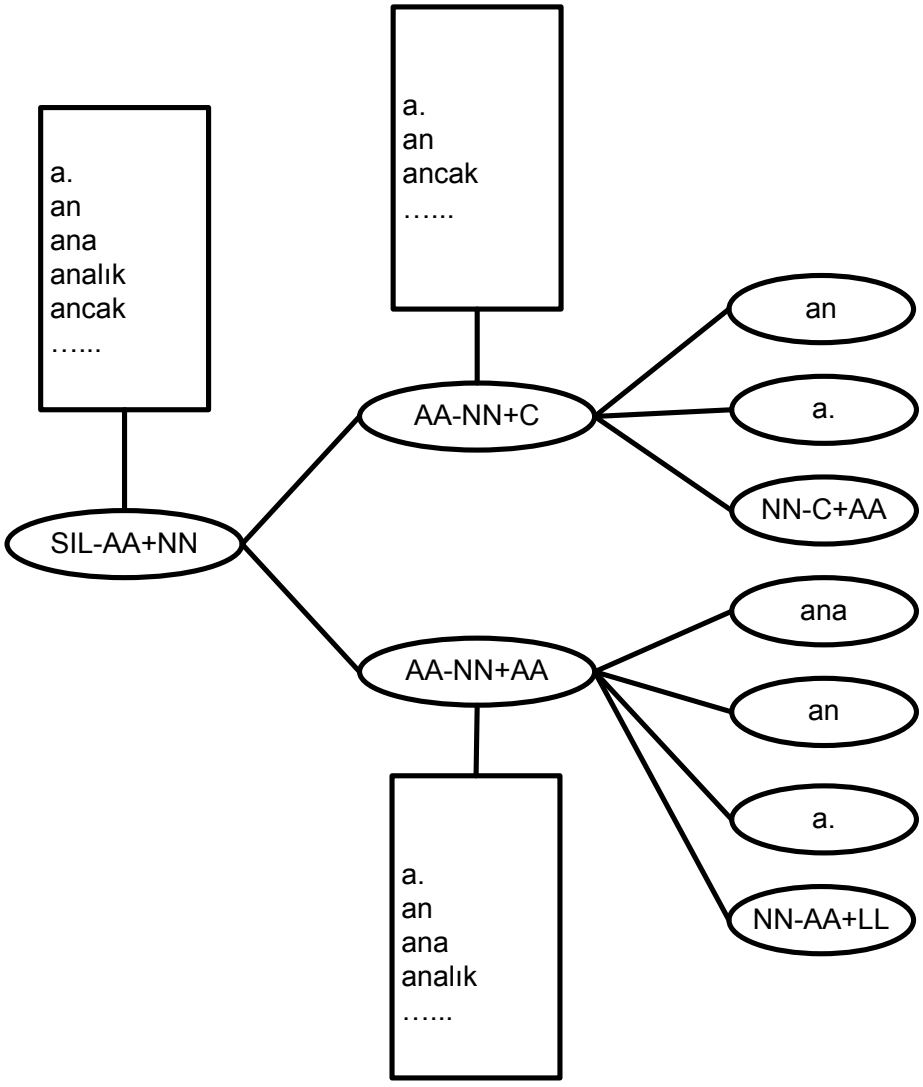
3.4.1 Çözümleme (Decoding)

Şimdiye kadar büyük-kelime hazineli sistemlerin ana kısımları ele alındı. Bu kısımları kullanarak tanıma yapmak için, Denklem 2.1'i maksimize eden $\hat{W}$ kelime sırasını bulunmalıdır.

Tüm arama sorunlarında olduğu gibi, iki ana yaklaşım vardır: Derinliğine arama (depth-first) ve enine arama (breadth-first). Derinliğine arama yönteminde en etkili hipotez konuşma sona erene dek takip edilir. Enine arama tasarımlarda, tüm hipotezler paralel takip edilir. Enine arama çözümlemesinde Bellman'ın optimality prensibini (Young, 1996) kullanlıır ve çoğunlukla Viterbi çözümleme olarak adlandırılır. Sürekli konuşma sistemleri karmaşık olduğu için, huzme arama (beam-search) adı verilen işlem kullanılmaktadır. HTK da huzme arama ve Viterbi çözümleme kullanır. Çözümleme sorununu anlamak için, her başlangıç kelimesine dallanan bir network ağacı düşünelim. Her ilk kelime takip edebilecek diğer kelimelere bağlıdır. Bu ağaç tabii ki çok büyük olacaktır. Az-kelimeli sistemlerde tüm kelimeleri paralel koyup aralarında döngü oluşturmak yeterlidir. Daha sonra her kelime yerine kelimenin telaffuzu yerleştirilir. Eğer birden fazla telaffuz var ise modeller paralel bağlanır.

Sonuç olarak durum geçişleri ile birbirine bağlı SMM-durum düğümleri ve kelime geçişleri ile birbirine bağlı kelime sonu düğümlerinde oluşan bir ağaç oluşur. Başlangıç düğümünden ağaçtaki her hangi bir noktaya gidiş, tüm log durum geçiş olasılıklarının, tüm log durum çıkış olasılıklar ve log-durum dil-model parametrelerinin toplanması ile değerlendirilir. Böyle bir yol, yolun sonuna hareketli bir işaret konarak temsil edilir. İşaretin o noktaya kadarki geçmişini gösteren bir değeri vardır. Yol mevcut işaretin takip eden başka bir düğüme geçmesi ve değerinin güncellenmesi ile uzatılabilir. Arama sorunu işaret-geçiş algoritması ile değiştirilebilir. İlk olarak başlangıç düğümüne bir işaret konulur. Her akustik vektör girdi yaptığında işaret yer değiştirir ve değeri güncellenir. Eğer aynı düğüme birden fazla işaret gelirse en iyi işaret Bellmans' optimality teorisine göre tespit edilir. Tüm akustik vektörler işlendikten sonra tüm kelime sonu düğümleri incelenir ve en yüksek değerli işarete sahip yol en muhtemel yol olduğu belirlenir. Bu temel işaret geçiş algoritması, en muhtemel yolu bulur fakat hesaplaması zordur. Bu metodu kolay işlenir hale getirmek

için budama (pruning) kullanılır. Her zaman birimi için, her hangi bir işaretteki en yüksek değer kaydedilir, bunun belli bir değer aşağısındakiler yok edilir. İşaretler ileriye devam ettikçe önde yeni bir ağaç yapısı oluşmakta, geride kalanlar ise yok edilmektedir. Bu işlemin çok verimli işlemesi için, işaretleri mümkün olan en kısa sürede budama çok önemlidir. Yolun sonuna gelene kadar kelimenin ne olduğu tam olarak anlaşılmaz. HTK dekoder, her işarete muhtemel kelimeyle ilişki kurar (Şekil 3.8). İşaret kelime sonuna ilerledikçe liste küçülür, tek kelimeye düşer.



Şekil 3.8 Dil modelinin öncel uygulamaları

3.4.2 Türkçe için yapılan diğer çalışmalar

Türkçe'ye özgü olarak Türkçenin Fince gibi sondan eklemeli bir dil olmasından hareketle morfem tabanlı tanıma sistemleri de kullanılmıştır (Çarkı K. et al. 2000). Uygulama olarak HTK kullanarak radyoloji alanında pratik uygulamalar da gerçekleştirilmiştir (Arısoy ve Arslan, 2004). Ayrıca yine HTK kullanılarak Türkçe Gazete Haberleri Dikte sistemi de geliştirilmiştir (Arısoy ve Arslan, 2005). Geniş dağarcıklı sürekli konuşma tanıma sistemi geliştirmek için HTK kullanarak hazırlanan yüksek lisans tezleri de vardır (Çömez, 2003; Şahin, 2003; Bayer, 2005). HTK dışında farklı Sonic (Pellom, 2001) gibi LVCSR tanıyıcılar da değişik çalışmalarda kullanılmıştır. Türkçe için konuşmacıdan geniş kelime hazneli sürekli kelime tanıma sistemi oluşturmak için tüm üçlü fonemleri içeren çok kapsamlı bir veritabanına ihtiyaç vardır. Ayrıca dil modeli oluşturmak amacıyla N-gram'ların ve üçlü seslerin ortaya çıkma sıklılarının bilinmesi gerekmektedir. Bunun için hem metin hem de ses kaydı içeren veritabanları oluşturulmalı; bu veri tabanlarından üçlü fonemleri tespit edilmeli; ve fonemler için SMM'ler oluşturulmadır. Mevcut sistemler (HTK, Sphinx, Sonic ve benzeri konuşma tanıma araçları) ile eğitim ve tanıtma işlemi gerçekleştirilebilir.

BÖLÜM 4

TÜRKÇENİN SES YAPISI

4.1 Türkçenin Ses Özelikleri

Dilin en küçük birimi fonem yani sestir. Sesin, oluşmasından itibaren ağızdan çıkana kadar izlediği güzergaha ses yolu denir. (Aynacı, M., 2007)'ye göre, ses telleri ile oluşturulan ses, gırtlak, küçük dil, dil, diş, damak, dudak gibi organların, ses yolunu açıp kapaması, daraltıp genişletmesi ile anlamlı seslere dönüşür ve konuşmayı oluşturur. Seslerin oluşumunda alt ve üst çenelerin de katkısı vardır. Türkçede sesler oluşumları bakımından ünlü ve ünsüz olmak üzere iki sınıfa ayrılır. Ses yolunda hiçbir engelle karşılaşmadan oluşan seslere ünlü, bir engelle karşılaşan seslere ise ünsüz denir.

4.1.1 Türkçede ünlüler ve ünsüzler

Türkçede sekiz ünlü vardır: a, e, ı, i, o, ö, u, ü. Ünlüler şöyle sınıflandırılır:

Oluşum noktalarına göre:

ön damak ünlüleri : e, i, ö, ü.

arka damak ünlüleri : a, ı, o, u.

Açıklık kapalılık derecelerine göre:

geniş ünlüler : a, e, o, ö.

dar ünlüler : ı, i, u, ü.

Dudakların durumuna göre:

düz ünlüler : a, e, ı, i.

yuvarlak ünlüler : o, ö, u, ü.

Türkçede ünsüzler yirmi bir tanedir: b, c, ç, d, f, g, ğ, h, j, k, l, m, n, p, r, s, ş, t, v, y, z. Ünsüzler de boğumlanma (oluşum) noktaları bakımından şöyle sınıflandırılabilir:

Dudak ünsüzleri: b, f, m, p, v.

Diş ve diş eti ünsüzleri: c, ç, d, l, n, r, s, ş, t, z.

Art damak ünsüzleri: ķ, ğ, l.

Ön damak ünsüzleri: k, g, l, y.

Gırtlak ünsüzleri: h

Ünsüzler ses tellerinin titreşip titreşmemesine göre ötümlü ünsüzler ve ötümsüz ünsüzler olarak ikiye ayrılır.

Ötümlü ünsüzler: b, c, d, g, ğ, l, m, n, r, v, y, z

Ötümsüz ünsüzler: ç, f, h, k, p, s, ş, t

Süreklilik ya da süreksizlik bakımından ünsüzler ise şöyledir:

Sürekli ünsüzler: f, ğ, h, j, l, m, n, r, s, ş, v, y, z

Süreksiz ünsüzler: b, c, ç, d, g, k, p, t

Ağız ya da geniz ünsüzü olma bakımından ünsüzler:

Geniz ünsüzleri : m, n ve Anadolu ağızlarında ñ

Ağız ünsüzleri : b, c, ç, d, f, g, ğ, h, j, k, l, p, r, s, ş, t, v, y, z.

4.1.2 Seslerin süreleri

Yapılan bir araştırmanın sonuçlarına göre seslerin sürelerinin birbirlerinden farklı olduğu ve bazı etkiler sonucunda da sürelerinin uzadığı belirlenmiştir. Bu etmenler söylenen ses, söylenen sesin çevresindeki sesler, söylenen sesin sözcük / öbek / tümce içindeki konumudur. (Şaylı, Ö ve Aslan L. M., 2003)'de , ünlülerin ortalama süreleri daha fazla olan geniş ünlüler ve ortalama süreleri kısa olan dar ünlüler diye iki sınıfa ayrıldığı; ötümsüz patlamalılar (/p/, /t/, /k/), ötümsüz sızmalılar (/f/, /s/, /ş/) ile /ç/, /j/ ve /z/ seslerinin ortalama sürelerinin diğer ünsüzlerinkine göre uzun olduğu ve ötümlü patlamaların (/b/, /d/, /g/), yarı ünlülerin (/r/, /y/, /l/), fısıltı sesinin (/h/) ve sürekli sızmalı sesinin (/v/) süreleri diğer ünsüzlere göre düşük olduğu belirtilir. Ünlüler açık, kapalı, kısa ve uzun olabilir. Uzun ya da kısa ünlüler günümüzde sesyazar (sonagramm) denen bir aygıtla saptanabilmektedir. (Selen, 1979)

Türkçe sözcüklerde uzun ünlü yoktur, ünlüler aslî uzunluğunu 'ya:d el', 'va:r ol' gibi sözcükler dışında yitirmiştir. Uzun ünlü dilimize yabancı dillerden giren sözcüklerde ve ünsüzlerin etkisiyle görülür. Sözcüklerdeki uzun ünlülerin kısalmasına ünlü kısalması, kısa ünlülerin uzamasına ise ünlü uzaması denir. Türkçede uzun ünlü bulunmadığından söyleyiş kolaylığını sağlamak için yabancı kökenli sözcüklerdeki uzun ünlüleri kısaltma eğilimi vardır. Bu eğilim Arapça ve Farsça kökenli sözcüklerde daha fazladır. Dilimizde çok eski dönemlerden bu yana kullanılan sözcüklerde ve halk dilinde kısalma daha sık görülür.

Örnekler:

a:vi:ze (Far.) > avi:ze	mevzua:t > mevzuat (Ar.)
meeting (İng.) > miting	kita:b > kitap (Ar.)
hesa:b (Ar.) > hesap	speaker > spiker (İng.)
mevcu:d (Ar.) > mevcut	mura:d (Ar.) > murat
hooligan (İng.) > holigan	ca:n (Far.) > can

4.2 Türkçedeki Fonem Yapısı

Günümüz Türkçesi, Fince ve Japonca gibi fonem tabanlı bir dildir. Fonem tabanlı dillerde fonemler harflerle temsil edilir. Yazılı metin ve telaffuzu arasında neredeyse birebir örtüşme vardır. Fakat bazı sesli ve sessiz harflerin ses yolunda (vocal tract) üretildiği yere bağlı olarak bazı farklılıkları vardır. Örneğin "laf" kelimesindeki "a" harfi ile "almak" kelimesindeki "a" farklı telaffuz edilir. Türkçe'deki "k" sessizi, "kasa" sözcüğünde arkadamak, "kedi" sözcüğünde öndamak olarak seslendirilmektedir (Ergenç, 2002). Bu nedenle Türkçe'deki ses özelliklerini ortaya çıkarmak için yazı dilinin fonetik sembollere çevrilmesi gerekmektedir. Bu nedenle Türkçe'deki 29 harf, konuşma dilindeki 43 farklı sese karşılık gelmektedir (Ergenç, 2002)

4.3 METUbet

ODTÜ tarafından (Ergenç, 2002)'deki fonetik sembol setine bağlı kalınarak yeni bir harf-fonem çevirim sistemi geliştirilmiştir. Bu sistem sözlüklerde yer alan kelimelerin fonetik karşılılarına ve yer aldıkları şartlara bağlı olarak geliştirilmiştir. Bu şartlar yer aldıkları kelime ve kelimedeki konumlarıdır. (Ergenç, 2002)'de yer alan IPA (International Phonetic Alphabet- Uluslararası Fonetik Abece) sembollerini kullanmak zor olduğundan Speech Assessment Method Phonetic Alphabet (SAMPA) sözlüğünde yer alan semboller (Ergenç, 2002)'deki sembollerle eşleştirilerek kullanılmıştır. SAMPA Türkçe'ye çok uygun olmadığı için eklemeler ve değiştirmeler yapılmış ve yeni bir basitleştirilmiş fonetik alfabe olan METUbet geliştirilmiştir. Sembol formatlama seçimleri Amerikan İngilizcesi için kullanılan ARPAbet ile benzerlik göstermektedir. SAMPA'nın 45 fonetik gösterimine kıyasla

METUbet'in 39 fonetik gösterimi vardır. Bunun nedeni ise u, ü, o, ö, ve i harflerinin uzun ve kısa telaffuzları tek bir fonem ile gösterilmesidir.

(Salor Ö. vd. 2002)'de bahsedilen METUbet Türkçedeki sesleri 39 farklı fonem ile gösterir. Çizelge 4.1'de diğer fonetik alfabeler ile olan karşılaştırılmaları görülmektedir.

Çizelge 4.1 METUbet'in diğer fonetik alfabelerle karşılaştırılması(Salor Ö. vd. 2002)

IPA	SAMPA	METUbet	Example
ɑ	A	AA	*a*nı
a	a	A	l*a*f
e	e	E	*e*lma
ɛ	E	EE	dere
i	i	IY	*i*ğde
ɪ	I	IY	sim*i*t
ï	1	I	*ı*sı
ɔ	O	O	s*o*ru
o	o	O	*o*ğlak
ʊ	U	U	k*u*lak
u	u	U	*u*ğ*u*r
œ	2	OE	*ö*rtü
ø	5	OE	*ö*ğ*r*en
ʏ	Y	UE	*ü*mit
y	y	UE	d*ü*ğme
b	b	B	*b*al
d	d	D	*de*de
g	G	GG	karga
ɟ	g	G	genç
h	h	H	*h*asta
ʒ	Z	J	müj*d*e
k	k	KK	a*k*ıl
c	c	K	kedi
l	L	L	*l*ey*l*ek
ɫ	l	LL	ku*l*
m	m	M	da*m*
n	n	NN	a*n*ı
ŋ	N	N	sü*n*gü
p	p	P	i*p*
r	r	R	*r*af
ɾ̃	R	RR	ı*r*mak
ɣ	4	RH	bi*r*
s	s	S	*s*e*s*
ʃ	S	SH	a*ş*ı
t	t	T	ü*t*ü
v	v	VV	*v*ar
ʋ	w	V	tavuk
j	j	Y	*y*at
:ɪ	yy	Y	huy
z	z	Z	a*z*ık
ʒ	zz	ZH	yo*z*
ʤ	DZ	C	*c*am
ʧ	TS	CH	se*ç*im
f	f	F	*f*asıl
:	:	GH	düğme
Sil	Sil	SIL	"silence"

BÖLÜM 5

SÜREKLİ KONUŞMA TANIMA SİSTEMİ GELİŞTİRME

Sürekli konuşma tanıma sistemleri için pratik bir uygulama yapmak için uygulamada kullanılabilecek tüm fonem ve üçlü fonemleri kapsayan bir veritabanına ihtiyaç vardır. Bu çalışmada sayı dizisi tanıma için SMM uygulaması yapıldı. Sayı tanıma peşpeşe gelen dört rakam veya üç haneli bir sayı olabilir. Bu bölümde sayı dizisi tanıma için yapılan işlemler adım adım anlatılmıştır.

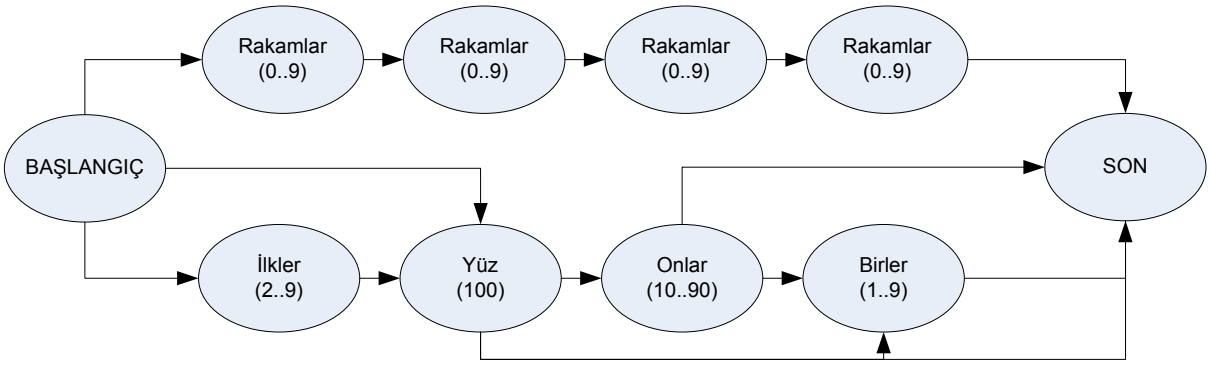
5.1 Gramerin Oluşturulması:

Uygulamaya yönelik bir gramer dosyası oluşturulması gerekir. HTK'nın anlayacağı ve kullanacağı şekilde aşağıdaki gramer dosyası oluşturulmuştur.

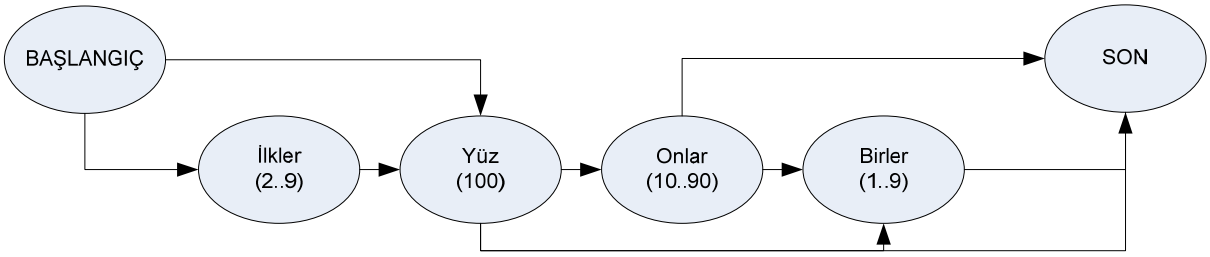
$sil=SIL;
$rakamlar= sIfIr | iki | UC | dOrt | beS | altI | yedi | sekiz | dokuz;
$ilkler= iki | UC | dOrt | beS | altI | yedi | sekiz | dokuz;
$birler= bir | iki | UC | dOrt | beS | altI | yedi | sekiz | dokuz;
$onlar= on | yirmi | otuz | kIrk | elli | altmIS | yetmiS | seksen | doksan;
$yzler= yUz;

($sil ((($ilkler | $sil) $yzler (($onlar $birler) | ($onlar) | ($birler) | ($sil))) | ($rakamlar $rakamlar $rakamlar $rakamlar)) $sil)

Kullanılan gramerin yapısı Şekil 5.1'de görülmektedir. Ayrıca bazı deneylerde kullanılan ve bu gramerin daha basitleştirilmiş şekli Şekil 5.2'de verilmiştir.



Şekil 5.1 Test grameri



Şekil 5.2 Basitleştirilmiş test grameri

5.2 Veritabanının Hazırlanması

Sürekli konuşma tanıma sistemleri için pratik bir uygulama geliştirmek amacıyla, uygulamada kullanılabilecek tüm fonem ve üçlü fonemleri kapsayan bir veritabanına ihtiyaç vardır. Belirli sayıda katılımcıdan uygulamaya yönelik konuşma verisi toplanmıştır. Veritabanı 36 erkek ve 15 kadın seslerinde oluşmaktadır. Konuşmacılar Çizelge 5.1'deki gibi gruplandırılmıştır. Eğitim dört grup üzerinden yapılıp, diğer grup test için kullanılmıştır. Deneysel çalışmada eğitim ve test beş farklı durum için tekrarlanarak ortalama başarımlar elde edilmiştir.

Çizelge 5.1 Deneylerde kullanılan veritabanının detayları

	Grup 1	Grup 2	Grup 3	Grup 4	Grup 5	Genel Toplam
Erkek	3	13	7	6	7	36
Kadın	2	3	3	4	3	15
Toplam	5	16	10	10	10	51

Veriler 16 kHz örnekleme frekansında alınmış ve daha sonra bir ses dosyası editörü ile parçalanarak WAV dosya formatı ile saklanmıştır. Ayrıca her ses dosyası için, eğitimde kullanılmak üzere etiket dosyaları oluşturulmuştur.

5.3 Eğitim İçin Ön Hazırlık:

1. HTK versiyon 3.4.1 kaynak kodları sitesinden indirilerek Linux Ubuntu 8.1 için derlenmiştir. Herhangi bir C derleyicisi bu iş için yeterlidir. HTK araçları (**HSLab** hariç) grafik arabirim kullanmaz. Bu nedenle çok fazla sistem belleği harcamaz. Ayrıca Windows tabanlı bilgisayarlar için de kolayca derlenebilir.

2. Uygulamadaki tüm sözcükleri içeren sözlük (*dict*) dosyası oluşturulmuştur.

3. Daha sonra aşağıda verilen klasörler oluşturulmuştur.

data/ : eğitim verilerini içerecek
data/train/ ve **data/test/** eğitim ve performans testi için ayrı verileri içerecek
analysis/ : akustik analiz safhası için gerekli dosyaları içerecek
training/ : başlangıç ve eğitim safhalarını içerek dosyaları içerecek.
model/ : tanıyıcının modelleri (SMM'ler)
def/ : işlemin tanım dosyalarını içerecek

4. METUbet'e uygun olarak SMMlist oluşturulmuştur.

AA	I	B	J	NN	RH	V	CH
A	O	D	KK	N	S	Y	F
E	U	GG	L	P	SH	Z	GH
EE	OE	G	LL	R	T	ZH	SIL
IY	UE	H	M	RR	VV	C	

5. MFCC parametrelerini oluşturmak için aşağıdaki seçenekleri içeren *config* ve *config2* dosyaları oluşturulmuştur. Ayrıca *yenilertargetlist* ve *alltestwordsphones.mlf* dosyaları da oluşturulmuştur. Dosya içerikleri EK Açıklamalar-A'da verilmiştir.

5.3.1 MFCC parametrelerini çıkarılması

Daha sonra aşağıdaki komut ile MFCC parametreleri oluşturulmuştur.

```
$> HCopy -A -D -C config -S yenilertargetlist -I alltestwordsphones.mlf
```

5.4 Eğitim

Ön hazırlıklar tamamlandıktan sonra fonemlerin eğitimine başlanabilir. Oluşturduğumuz veri tabanındaki etiket dosyalarında fonemlerin sınır değerleri belli olmadığı için **HInit** ve **HRest** araçlarını kullanamayacaktır. Bunun yerine tüm SMM'lere başlangıçta ortak bir değer atayan **HCompV** ile düz (flat) başlangıç yapılır.

5.4.1 Fonemlerin eğitimi

1. Prototip SMM dosyası olarak *proto* isimli dosya oluşturulmuştur. Dosyanın içeriği de EK Açıklamalar-A'da verilmiştir.

2. Daha sonra aşağıdaki komut ile **HCompV** aracı kullanılarak ile düz başlangıç yapılmıştır.

```
$> HCompV -C config2 -f 0.01 -m -S trainlistYeni -M model1/hmm0 proto
```

Dosya içindeki değerler düz başlangıç ile ortak bir ortalama ve varyans değerine ulaşmıştır. Kıyaslama için bu dosya da EK Açıklamalar-A'da verilmiştir. Elde edilen bu yeni *proto* dosyası tüm fonemler için teker teker çoğaltılarak *hmmdefs* ve *macros* dosyaları oluşturulmuştur.

3. HERest ile üst üste eğitimler yapılmıştır.

```
HERest -A -D -T 1 -C config2 -I Yeniphones0.mlf -t 250.0 150.0 1000.0 -S trainlistYeni \
  -H model1/hmm0/macros -H model1/hmm0/hmmdefs -M model1/hmm1 monophones0

HERest -A -D -T 1 -C config2 -I Yeniphones0.mlf -t 250.0 150.0 1000.0 -S trainlistYeni \
  -H model1/hmm1/macros -H model1/hmm1/hmmdefs -M model1/hmm2 monophones0

HERest -A -D -T 1 -C config2 -I Yeniphones0.mlf -t 250.0 150.0 1000.0 -S trainlistYeni \
  -H model1/hmm2/macros -H model1/hmm2/hmmdefs -M model1/hmm3 monophones0
```

4. Bir sonraki aşama olarak *sp* (short pause- kısa duraklama) adı verilen yeni SMM, SIL SMM'sinin orta durum geçiş değerleri kopyalanarak oluşturulmuştur.

5. **HHed** ve **HVite** araçları ile etiket dosyaları *sp* fonemini içerecek şekilde tekrar düzenlenmiş ve **HERest** ile birkaç kere daha eğitim yapılmıştır.

5.4.2 Üçlü fonemlerin (trifon) eğitimi

Fonemlerin teker teker eğitilmesini takiben, etiket dosyalarının üçlü fonemlere göre düzenlenmesi ve üçlü fonemlerin eğitilmesi gereklidir. Aşağıda görülen işlem basamakları ile üçlü fonem eğitime devam edildi..

1. HLEd komutu kullanılarak veritabanında yer alan tüm üçlü fonem grupları listelenir ve etiket dosyaları bunlara uygun olarak değiştirilir.

```
$> HLEd -n triphones1 -l '*' -i Yeniwintri.mlf mktri.led Yenialigned.mlf
```

2. HHEd ile bu *hmmdefs* dosyası üçlü fonemler de ilave edilerek güncellenir. Veritabanında üçlü fonem tekrarı az olduğundan a-b+c gibi bir üçlü foneme b tekli fonemindeki veriler kopyalanır.

```
$> HHEd -H model1/hmm9/macros -H model1/hmm9/hmmdefs -M model1/hmm10
mktri.hed monophones1
```

3. HERest ile birkaç eğitim daha yapılır.

```
$> HERest -A -D -T 1 -C config2 -I Yeniwintri.mlf -t 250.0 150.0 1000.0 -s stats -S
trainlistYeni -H model1/hmm10/macros -H model1/hmm10/hmmdefs -M model1/hmm11
triphones1


$> HERest -A -D -T 1 -C config2 -I Yeniwintri.mlf -t 250.0 150.0 1000.0 -s stats -S
trainlistYeni  -H model1/hmm11/macros -H model1/hmm11/hmmdefs -M model1/hmm12
triphones1
```

4. Daha sonra durum bağlama işlemi için gerekli olan *tree.hed* isimli dosya oluşturulur. *mktri.hed* ve *tree.hed* dosyalarının içerikleri Ek Açıklamalar-B'de verilmiştir.

5. Bir sonraki aşamaya geçmeden hem tekli fonemleri hem de üçlü fonemleri içeren *fulllist* isimli dosya oluşturulur.

6. Aşağıdaki komut ile tiedlist adı verilen durumları bağlanmış fonem listesi oluşturulur.

```
$> HHEd -H model1/hmm12/macros -H model1/hmm12/hmmdefs -M model1/hmm13
tree.hed triphones1 > log
```

7. HERest ile birkaç eğitim daha yapılır.

```
$> HERest -A -D -T 1 -C config2 -I Yeniwintri.mlf -t 250.0 150.0 1000.0 -s stats -S
trainlistYeni -H model1/hmm13/macros -H model1/hmm13/hmmdefs -M model1/hmm14
tiedlist

$>  HERest -A -D -T 1 -C config2 -I Yeniwintri.mlf -t 250.0 150.0 1000.0 -s stats -S
trainlistYeni -H model1/hmm14/macros -H model1/hmm14/hmmdefs -M model1/hmm15
tiedlist
```

8. Artık eğitim tamamlanmıştır. Bir sonraki aşama olan tanıma işlemine geçilebilir.

5.5 Rakam Dizisi Tanıma

Telaffuz edilen üç basamaklı (100-999 arası) sayı veya dört sıralı rakamın (0000-9999 arası) tanınması için önceden veritabanındaki veriler kullanılmıştır. Şekil 5.1 verilen gramer kullanılmıştır. Ayrıca sadece son deney için şekil 5.2'de verilen basit gramer kullanılmıştır. **HVite** komutu ile tanıma gerçekleştirilmiş ve **HResults** ile referans dosyalar ile kıyas yapılarak tanıma oranları çıkarılmıştır. Örnek **HVite** komutu aşağıdadır.

```
$>  HVite -H model1/hmm15/macros -H model1/hmm15/hmmdefs -S test.scp -l '*' -i
recout07062010_1.mlf -w wdnet -p 25.0 -s 30.0 newdictMETU tiedlist
```

Bununla beraber kullanılan **HResults** komutu da aşağıdadır.

```
$>  HResults -I refwords.mlf tiedlist recout07062010_1.mlf
```

HResults'ın aşağıdaki gibi bir ekran çıktısı oluşur.

```
====================== HTK Results Analysis =======================
        Date: Mon Jun  7 23:22:29 2010
        Ref : refwords.mlf
        Rec : recout07062010_1.mlf
        ---------------------- Overall Results --------------------------
        SENT: %Correct=96.00 [H=48, S=2, N=50]
        WORD: %Corr=100.00, Acc=98.97 [H=194, D=0, S=0, I=2, N=194]
============================================================
```

HResults'ın ekran çıktısında SENT gramerdeki cümleyi göstermektedir. Toplam tanınması gereken 50 cümle olup, bunun 48 tanesi doğru tanınmış ve doğruluk oranı %96'dır. WORD ise cümlelerdeki sözcükleri göstermektedir. Cümlelerde yer alan 194 kelime % 100 oranında tanınmış, ancak iki adet kelime ilave edilmiştir. Cümleler teker teker gözden geçirildiğinde bu ilavelerin cümle başında olduğu görülmektedir. İlave kelime içeren bir örnek aşağıda görülmektedir. Ses verisindeki gürültüden dolayı cümlenin başına UC kelimesi eklenmiştir.

"/197.rec"*
300000 1100000 UC -1008.724121
1100000 2300000 yUz -1281.067505
2300000 7200000 doksan -5232.896973
7200000 12000000 yedi -4766.713867
.

Deneylerde ortaya çıkan bir başka durum ise kelime yerine başka bir kelimenin atanmasıdır. Bununla ilgili bir örnek de aşağıda verilmiştir.

"/351.rec"*
500000 3200000 iki -3109.456543
3200000 4700000 yUz -1846.029053
4700000 6600000 elli -2134.210449
6600000 9400000 bir -3108.593262
.

Daha önce bahsettiğimiz beş (5) ayrı grup aşağıda görüldüğü gibi değişik şekillerde eğitime katılmış, eğitime katılmayan diğer gruba ait cümleler test için kullanılmıştır. Herhangi bir kelimenin yanlış tanınması, görülememesi cümlenin yanlış tanınmasına sebep olmaktadır. Bu nedenle deneylerde cümle tanıma oranları kelime tanıma oranlarından düşük çıkmıştır. Deney sonuçları Çizelge 5.1'de verilmiştir. **HResults** çıktısında görülen kısaltmaların açıklaması aşağıda verilmektedir.

H: Doğru tanınan
D: (Deletion) Görülmeyen
S: (Substitution) Yerine atama
I: (Insertion) İlave kelime
N: Toplam sayı

Çizelge 5.2 Yapılan deneylerin sonuçları

EğitimGrupları	**Test Grubu**		**Tanıma Oranları (%)**	**H**	**D**	**S**	**I**	**N**
Grup 1, 2, 3, 4	Grup 5	Cümle	71,93	123	-	48	-	171
		Kelime	91,95	605	5	48	1	658
Grup 1, 2, 3, 5	Grup 4	Cümle	75,29	128	-	42		170
		Kelime	92,34	603	6	44	3	653
Grup 1, 2, 4, 5	Grup 3	Cümle	73,37	124	-	45	-	169
		Kelime	92	598	5	47	3	650
Grup 1, 3, 4, 5	Grup 2	Cümle	73,18	191	-	70	-	261
		Kelime	91,17	929	13	77	10	1019

EğitimGrupları	Test Grubu		Tanıma Oranları (%)	H	D	S	I	N
Grup 2, 3, 4, 5	Grup 1	Cümle	93,75	75	-	5	-	80
		Kelime	99,04	311	0	3	2	314

Deneysel çalışma sonunda ortalama cümle tanıma oranı %73,5 ve kelime tanıma oranı %92,24 olarak elde edilmiştir. Fonetik benzerliği olan kelimelerde görülen hatalı tanıma cümle tanıma başarımlarını düşürmektedir. Bunun da nedeni üçlü fonem modellerinin yeteri kadar iyi eğitilmemesinden kaynaklanmaktadır. Daha büyük veritabanı ve fonem sınırları belli etiket dosyaları kullanılarak sistem başarımının iyileştirilebileceği söylenebilir. Bu çalışmaların yanında Şekil 5.2'de verilen basit gramer kullanılarak da deney yapılmıştır. Bunun için Grup 2,3,4 ve 5 eğitim ve Grup 1 test işleminde kullanılmıştır. Yapılan deney sonuçları aşağıdaki Çizelge 5.2'de verilmiştir. Deney sonunda cümle ve kelime tanıma oranları sırasıyla %96 ve 100 olarak elde edilmiştir.

Çizelge 5.3 Basit gramer kullanılarak elde edilen tanıma oranları

EğitimGrupları	Test Grubu		Tanıma Oranları (%)	H	D	S	I	N
Grup 2, 3, 4, 5	Grup 1 (Basit Gramer)	Cümle	96	48	-	2	-	50
		Kelime	100	194	0	0	2	194

5.6 Değerlendirme

Sonuçlar incelendiğinde bazı kişilerdeki yanlış tanıma oranlarının yüksek olduğu görülmüştür. Veritabanı yeteri kadar büyük olmadığı için farklı şöyleyişler ve telafuzlar içindeki üçlü fonemler çok iyi modellenmemektedir. Başarımı yükseltmek için veritabanındaki kişi sayısı arttırılmalıdır. Ayrıca fonetik benzerlikten dolayı yedi-iki ve beş-bir (bazı kişilerde bilhassa) sözcüklerinde yanlış atamalar olmaktadır. Eğitimde **HCompV** kullanarak tüm fonem SMM'leri genel bir ortalama ve varyans değeri ile ortak başlangıç ile değeri atanmıştır (flat start). Sınır değerleri belli etiket dosyaları kullanarak HInit ve HRest

komutları ile eğitime başlanırsa, SMM'lerin başlangıç değerleri daha doğru olacak ve tanıma oranları da artacaktır.

HTK, tanıma işleminin gerçek zamanlı yapılabilmesine olanak sağlamaktadır. Bu çalışmada tanıma sistemini gerçek zamanlı çalıştırılmış ve sistemin başarımı gözlenmiştir. HTK ile eğitim bir çok ön hazırlık gerektirmesine ve birçok farklı dosyaya ihtiyaç duyulmasına rağmen tanıma işlemi görece daha basittir. Gerçek zamanlı tanımada kullanılan **HVite** aracı, eğitim sonucu elde edilen ve tüm SMM'leri içeren iki dosyaya (*hmmdefs* ve *macros* dosyaları); tüm fonem, üçlü fonem ve durum bağlama sonucu elde edilen grupleri içeren dosyaya (*tiedlist*); ve uygulamadaki tüm kelimeleri içeren bir sözlük (*dict*) dosyasına ihtiyaç duyar. Yaptığımız uygulamada bu dosyaların toplam boyutu 1 MByte'dan az olduğu görülmüştür. HTK'nın kullandığı diğer bellek miktarı da fazla değildir. HTK, grafiksel işlem yapılmadığı için, herhangi bir Linux veya Windows tabanlı bilgisayarda çalıştırılabilir. Uygulamanın karmaşıklığına göre ihtiyaç duyacağı bellek ve işlem süresi değişebilecektir.

BÖLÜM 6

SONUÇ VE ÖNERİLER

Bu tez çalışmasında Türkçe sürekli konuşma tanıma ve saklı Markov modeller (SMM) üzerinde çalışılmıştır. SMM ile uygulama geliştirme ve araştırma yapmak için kullanılan araçlar mevcuttur. Bunlardan Hidden Markov Tool Kit (HTK) ile tez çalışmasında rakam dizisi tanıma uygulaması geliştirilmiştir. Tez çalışmasının temel amacı Eskişehir Osmangazi Üniversitesi'nde konuşma tanıma çalışmalarında Türkçe sürekli ses tanıma için gerekli bilgi birikimi sağlamaktır. Bunun için rakam dizisi tanım uygulaması seçilmiş ve buradan gerekli bilgi ve deneyimler elde edilmiştir.

Rakam dizisi tanıma uygulamasında üç rakamlı sayı ve rakam (ardı ardına gelen dört farklı rakam) dizisi tanıma yapılmıştır. Bu uygulama, sayılarla ilgili herhangi bir pratik uygulama için ön çalışma olabilir. Öğrenci notları veya sayısal şifreler sesli komutlar bilgisayara aktarılabilir. Buna ilave olarak engelli veya iki eli de meşgul birinin okuduğu bir değerin bilgisayara aktarılması için de kullanılabilir. Henüz geliştirme aşamaları devam eden F-35 gibi yeni nesil savaş uçaklarında bir eli gaz kolu, diğer eli lövyede olan pilotun haberleşme veya seyrüsefer frekansı gibi sayısal değerleri ses komutuyla seçeceği değişik kaynaklarda belirtilmektedir. Ayrıca, Türkçe'de insanlar telefon numarası, kimlik numarası, vergi numarası, sosyal güvenlik numarası gibi uzun rakam dizilerini genelde üçer üçer ve/veya ikişer ikişer okurlar. Bu tip sayı dizilerinin sesli olarak bilgisayara girilebilmesi için de bu uygulama ile eğitilen SMM'ler kullanılabilecektir.

Yapılan deneysel çalışmada HTK tanıma sisteminin istenildiği gibi çalıştığı görülmüştür. Sistem, rakam dizisi tanıma yanında gramer değiştirilerek farklı uygulamalar için de kullanılabilir. Deneysel çalışmada elli kişini ses verilerinden oluşan bir veri tabanı kullanılmıştır. Sistemin rakam dizilerini tanınma oranı %74 civarındadır. Bu sonucun düşük çıkmasının nedeni veritabanının yeteri kadar büyük olmamasındandır. Sistemin başarımını arttırmak için veritabanındaki kişi sayısı arttırılmalı ve etiketlenmiş ses verileri üzerinde model eğitimi yapılmalıdır. Karışan sözcük modelleri için birden fazla SMM kullanılabilir. Ayrıca sistem başarımını arttırmak için farklı öznitelikler kullanılabilir.

Ek Açıklamalar A

HTK'da KULLANILAN DOSYALARIN İÇERİKLERİ

config:

```
SOURCEKIND=WAVEFORM
SOURCEFORMAT=WAV
SOURCERATE=625
TARGETKIND=MFCC_0_D_A
USEHAMMING=T
SAVEWITHCRC=T
SAVECOMPRESSED=T
PREEMCOEF=0.97
ENORMALISE=F
TARGETRATE=100000.0
WINDOWSIZE=250000.0
CEPLIFTER=22
NUMCHANS=26
NUMCEPS=12
USEPOWER=T
```

config2:

```
TARGETKIND=MFCC_0_D_A
USEHAMMING=T
SAVEWITHCRC=T
SAVECOMPRESSED=T
PREEMCOEF=0.97
ENORMALISE=F
TARGETRATE=100000.0
```

WINDOWSIZE=250000.0
CEPLIFTER=22
NUMCHANS=26
NUMCEPS=12
USEPOWER=T

yenilertargetlist:

./kayıtlar/yeniler/e10/1244.wav ./kayıtlar/yeniler/e10/1244.mfcc
./kayıtlar/yeniler/e10/126.wav ./kayıtlar/yeniler/e10/126.mfcc
./kayıtlar/yeniler/e10/2543.wav ./kayıtlar/yeniler/e10/2543.mfcc
./kayıtlar/yeniler/e10/268.wav ./kayıtlar/yeniler/e10/268.mfcc
.
.

alltestwordsphones.mlf **:**

"*/134.lab"
SIL
Y
UE
ZH
O
T
U
ZH
D
OE
RR
T
SIL
.

"*/135.lab"
SIL
Y
UE
ZH
.
.

proto:

~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"
<BeginHMM>
<NumStates> 5
<State> 2
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 3
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0

<State> 4
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<TransP> 5
0.0 1.0 0.0 0.0 0.0
0.0 0.6 0.4 0.0 0.0
0.0 0.0 0.6 0.4 0.0
0.0 0.0 0.0 0.7 0.3
0.0 0.0 0.0 0.0 0.0
<EndHMM>

HCompV ile yapılan flat-start sonucu oluşan yeni **proto** dosyası:

~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_D_A_0><DIAGC>
~h "proto"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 39
-1.236158e+01 -9.986625e+00 2.945845e+01 -2.605437e+01 1.245057e+01 -2.842183e+01
2.232224e+00 -7.659791e+00 -1.570798e+00 1.855082e+00 -3.091699e-01 4.107651e+00
1.028352e+02 -2.860786e-02 -2.738349e-02 2.974345e-02 -2.904162e-02 -2.077013e-02 -
4.804805e-02 1.691578e-03 6.554352e-02 6.571160e-02 5.847974e-02 3.444790e-02
5.270095e-02 5.372359e-03 -2.669566e-04 6.423678e-03 -2.139572e-02 2.425863e-02 -

4.162039e-03 2.801440e-02 2.707456e-03 7.306456e-03 4.206609e-03 3.051188e-03
8.981505e-03 6.574846e-04 -2.409511e-02
<VARIANCE> 39
3.886910e+02 2.575576e+02 6.048333e+02 5.993515e+02 3.998059e+02 6.926826e+02
4.215773e+02 3.647601e+02 3.706738e+02 2.916676e+02 2.402004e+02 1.803908e+02
6.088458e+02 1.338065e+01 1.112986e+01 1.734534e+01 2.214137e+01 1.838627e+01
2.259381e+01 1.826170e+01 1.638753e+01 1.574410e+01 1.328197e+01 1.278557e+01
9.445745e+00 1.634534e+01 1.756968e+00 1.659962e+00 2.425489e+00 3.122137e+00
2.920912e+00 3.221176e+00 2.969968e+00 2.597526e+00 2.521123e+00 2.191443e+00
2.135625e+00 1.594907e+00 1.662489e+00
<GCONST> 1.955843e+02
<STATE> 3
<MEAN> 39
-1.236158e+01 -9.986625e+00 2.945845e+01 -2.605437e+01 1.245057e+01 -2.842183e+01
2.232224e+00 -7.659791e+00 -1.570798e+00 1.855082e+00 -3.091699e-01 4.107651e+00
1.028352e+02 -2.860786e-02 -2.738349e-02 2.974345e-02 -2.904162e-02 -2.077013e-02 -
4.804805e-02 1.691578e-03 6.554352e-02 6.571160e-02 5.847974e-02 3.444790e-02
5.270095e-02 5.372359e-03 -2.669566e-04 6.423678e-03 -2.139572e-02 2.425863e-02 -
4.162039e-03 2.801440e-02 2.707456e-03 7.306456e-03 4.206609e-03 3.051188e-03
8.981505e-03 6.574846e-04 -2.409511e-02
<VARIANCE> 39
3.886910e+02 2.575576e+02 6.048333e+02 5.993515e+02 3.998059e+02 6.926826e+02
4.215773e+02 3.647601e+02 3.706738e+02 2.916676e+02 2.402004e+02 1.803908e+02
6.088458e+02 1.338065e+01 1.112986e+01 1.734534e+01 2.214137e+01 1.838627e+01
2.259381e+01 1.826170e+01 1.638753e+01 1.574410e+01 1.328197e+01 1.278557e+01
9.445745e+00 1.634534e+01 1.756968e+00 1.659962e+00 2.425489e+00 3.122137e+00
2.920912e+00 3.221176e+00 2.969968e+00 2.597526e+00 2.521123e+00 2.191443e+00
2.135625e+00 1.594907e+00 1.662489e+00
<GCONST> 1.955843e+02
<STATE> 4
<MEAN> 39

-1.236158e+01 -9.986625e+00 2.945845e+01 -2.605437e+01 1.245057e+01 -2.842183e+01
2.232224e+00 -7.659791e+00 -1.570798e+00 1.855082e+00 -3.091699e-01 4.107651e+00
1.028352e+02 -2.860786e-02 -2.738349e-02 2.974345e-02 -2.904162e-02 -2.077013e-02 -
4.804805e-02 1.691578e-03 6.554352e-02 6.571160e-02 5.847974e-02 3.444790e-02
5.270095e-02 5.372359e-03 -2.669566e-04 6.423678e-03 -2.139572e-02 2.425863e-02 -
4.162039e-03 2.801440e-02 2.707456e-03 7.306456e-03 4.206609e-03 3.051188e-03
8.981505e-03 6.574846e-04 -2.409511e-02
<VARIANCE> 39
3.886910e+02 2.575576e+02 6.048333e+02 5.993515e+02 3.998059e+02 6.926826e+02
4.215773e+02 3.647601e+02 3.706738e+02 2.916676e+02 2.402004e+02 1.803908e+02
6.088458e+02 1.338065e+01 1.112986e+01 1.734534e+01 2.214137e+01 1.838627e+01
2.259381e+01 1.826170e+01 1.638753e+01 1.574410e+01 1.328197e+01 1.278557e+01
9.445745e+00 1.634534e+01 1.756968e+00 1.659962e+00 2.425489e+00 3.122137e+00
2.920912e+00 3.221176e+00 2.969968e+00 2.597526e+00 2.521123e+00 2.191443e+00
2.135625e+00 1.594907e+00 1.662489e+00
<GCONST> 1.955843e+02
<TRANSP> 5
0.000000e+00 1.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
0.000000e+00 6.000000e-01 4.000000e-01 0.000000e+00 0.000000e+00
0.000000e+00 0.000000e+00 6.000000e-01 4.000000e-01 0.000000e+00
0.000000e+00 0.000000e+00 0.000000e+00 7.000000e-01 3.000000e-01
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
<ENDHMM>

Ek Açıklamalar B

ÜÇLÜ FONEM EĞİTİMLERİNDE KULLANILAN DOSYALARIN İÇERİKLERİ

mktri.hed:

```
CL triphones1
TI T_AA {(*-AA+*,AA+*,*-AA).transP}
TI T_A {(*-A+*,A+*,*-A).transP}
TI T_E {(*-E+*,E+*,*-E).transP}
….
.
.
TI T_V {(*-V+*,V+*,*-V).transP}
TI T_Y {(*-Y+*,Y+*,*-Y).transP}
TI T_Z {(*-Z+*,Z+*,*-Z).transP}
TI T_ZH {(*-ZH+*,ZH+*,*-ZH).transP}
TI T_C {(*-C+*,C+*,*-C).transP}
TI T_CH {(*-CH+*,CH+*,*-CH).transP}
TI T_F {(*-F+*,F+*,*-F).transP}
TI T_SIL {(*-SIL+*,SIL+*,*-SIL).transP}
TI T_sp {(*-sp+*,sp+*,*-sp).transP}
```

tree.hed:

```
RO  100.0 stats
TR  0
QS  "L_v_ince_dar" {IY-*,E-*}
QS  "R_v_ince_dar" {*+IY,*+E}
QS  "L_v_ince_yuv" {OE-*,UE-*}
```

```
QS  "R_v_ince_yuv" {*+OE,*+UE}
QS  "L_v_kalin_dar" {A-*,AA-*,I-*,EE-*}
.
.
.
QS  "L_unv_stop" {T-*,P-*,CH-*}
QS  "R_unv_stop" {*+T,*+P,*+CH}
QS  "L_unv_stop1" {KK-*}
QS  "R_unv_stop1" {*+KK}
.
.
.
QS  "R_diger1" {*+R,*+RR,RH-*}

QS  "L_A" {A-*}
QS  "R_A" {*+A}
QS  "L_AA" {AA-*}
QS  "R_AA" {*+AA}
.
.
.
QS  "L_ZH" {ZH-*}
QS  "R_ZH" {*+ZH}
QS  "L_SIL" {SIL-*}
QS  "R_SIL" {*+SIL}
QS  "L_sp" {sp-*}
QS  "R_sp" {*+sp}
TR 2
TB 350 "ST_AA_2_" {("AA","*-AA+*","AA+*","*-AA").state[2]}
TB 350 "ST_A_2_" {("A","*-A+*","A+*","*-A").state[2]}
TB 350 "ST_E_2_" {("E","*-E+*","E+*","*-E").state[2]}
TB 350 "ST_EE_2_" {("EE","*-EE+*","EE+*","*-EE").state[2]}
```

```
.
.
.
TB 350 "ST_OE_3_" {("OE","*-OE+*","OE+*","*-OE").state[3]}
TB 350 "ST_UE_3_" {("UE","*-UE+*","UE+*","*-UE").state[3]}
TB 350 "ST_B_3_" {("B","*-B+*","B+*","*-B").state[3]}
TB 350 "ST_D_3_" {("D","*-D+*","D+*","*-D").state[3]}
TB 350 "ST_GG_3_" {("GG","*-GG+*","GG+*","*-GG").state[3]}
TB 350 "ST_G_3_" {("G","*-G+*","G+*","*-G").state[3]}
.
.
.
TB 350 "ST_GG_4_" {("GG","*-GG+*","GG+*","*-GG").state[4]}
TB 350 "ST_G_4_" {("G","*-G+*","G+*","*-G").state[4]}
TB 350 "ST_H_4_" {("H","*-H+*","H+*","*-H").state[4]}
TB 350 "ST_J_4_" {("J","*-J+*","J+*","*-J").state[4]}
TB 350 "ST_K_4_" {("K","*-K+*","K+*","*-K").state[4]}
.
.
.
TB 350 "ST_CH_4_" {("CH","*-CH+*","CH+*","*-CH").state[4]}
TB 350 "ST_F_4_" {("F","*-F+*","F+*","*-F").state[4]}
TB 350 "ST_SIL_4_" {("SIL","*-SIL+*","SIL+*","*-SIL").state[4]}
TB 350 "ST_sp_4_" {("sp","*-sp+*","sp+*","*-sp").state[4]}

AU "fulllist"
CO "tiedlist"
ST "trees"
```

KAYNAKLAR DİZİNİ

Arısoy E., Arslan L.M., 2004, Turkish Radiology Dictatiton System SPECOM'2004: 9th Conference Speech and Computer St. Petersburg, Russia September 20-22

Arısoy, E. ve Arslan, L.M.,2005,Turkish Dictation Sytem for Broadcast News Applications,Signal Processing and Communications Applications Conference, Proceedings of the IEEE 13th Volume , Issue , 629 – 632

Arslan, L. M., 1999, Konuşma Tanıma ve Konuşma Sentezi Uygulamalarında en Uygun Fonetik Dizgenin Otomatik Seçimi Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU-1999), Denizli, Türkiye, 539-542.

Aynacı, M., 2007, Türkiye Türkçesinde Ses Etkileşimleri, Kocaeli Üniversitesi, Yüksek Lisans Tezi.

Bayer, A. O., 2005, "A study on language modeling for Turkish large vocabulary continuous speech recognition," Master's thesis, Middle East Technical University, Turkey.

Çarkı K., Geutner P., and Schultz T., 2000 "Turkish LVCSR: Towards better speech recognition for agglutinative languages," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, 1563–1566, June 2000.

Çömez, M. A., 2003, "Large vocabulary continuous speech recognition for Turkish using HTK," Master's thesis, Middle East Technical University, Turkey.

Dalkılıç, G. Çebi, Y., 2004,. Word Statistics of Turkish Language on a Large Scale Text Corpus – TurCo., Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), Volume 2 - Volume 2.

Davis, S. B. and Mermelstein, P., 1980, Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences, IEEE Trans. ASSP, Vol 28, 357–366.

DİKTE-Türkçe Sürekli Konuşma Tanıma Sistemi (Son Erişim tarihi Temmuz 2010) http://www.dikte.com.tr

Edizkan, R., 1999. "Gizli Markov Model ile Bilgisayarda Konuşma Tanıma: Özellik Uzayında ve Altuzayda Sınıflandırıcı Tasarımı., Eskişehir Osman Gazi Üniversitesi, Doktora Tezi.

KAYNAKLAR DİZİNİ (devam ediyor)

Ergenç, İ., 2002, Spoken Language and Dictionary of Turkish Articulation, İstanbul 486 p.

HTK-Hidden Markov Tool-Kit (Son Erişim tarihi Temmuz 2010)
http://htk.eng.cam.ac.uk

Huang, X. Acero A., Hon H. W., 2001, Spoken Language Processing: A guide to Theory, Algorithm, and System Development, Prentice Hall, Englewood Cliffs, NJ, 1008 p.

Lee K. F. , Hon H. W., Reddy R., 1990, An Overview of the SPHINX Speech Recognition System, IEEE Trans. On Acoustics, Speech and Signal Processing, Vol.38, No.1, January 1990, p: 35-45.

Pellom, B. "Sonic:The University of Colorado Continuous Speech Recognizer", Technical Report TR-CLSR-2001-01 CLSR, University of Colorado, March 2001.

Quatieri, T. F., 2002, Discrete Time Speech Signal Processing. Prentice Hall, Upper Saddle River, NJ, 816 p.

Rabiner L., 1989 A Tutorial On Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, Vol.77, No.2, February 1989, 257-289.

Rabiner L. and Juang B. H., 1993. Fundamentals of Speech Recognition, Prentice Hall, Englewood Cliffs, NJ, 496 p.

Salor Ö., Pellom B., Çiloğlu T., Hacıoğlu K., Demirekler M., 2002 On developing new text and audio corpora and speech recognition tools for the Turkish language, in Proceedings of 7th International Conference on Spoken Language Processing, September 2002, 349–352.

Sampa for Turkish (Son Erişim tarihi Temmuz 2010)
http://www.phon.ucl.ac.uk/home/sampa/turkish.htm

SESTEK-Türkçe Sürekli Konuşma Tanıma Sistemi (Son Erişim tarihi Temmuz 2010)
http://www.sestek.com.tr

Şahin S., 2003, "Language modeling for Turkish continuous speech recognition," Master's thesis, Middle East Technical University, Turkey.

Şayli Ö. ve Arslan L. M., 2003, Türkçedeki Seslerin Süre Özellikleri", Dilbilim Araştırmaları İstanbul 2003, 15-16.

Selen, N. 1979, Söyleyiş Sesbilimi Akustik Sesbilim ve Türkiye Türkçesi, Ankara, Türk Dil Kurumu Yayınları, 131 s.

KAYNAKLAR DİZİNİ (devam ediyor)

Young, S., 1996, A review of large-vocabulary continuous-speech recognition, IEEE Signal Processing Magazine, vol 13, no.5,.45-57

Young S., Evermann G., Gales M., Hain T., Kershaw D., Liu X., Moore G., Odell J., Ollason D., Povey D., Valtchev V., Woodland P., 2009, The HTK book (for HTK Version 3.4)", Cambridge University Engineering Department, 384 p.