KONUŞMA TANIMA TEKNOLOJİSİ YARDIMIYLA İLKÖĞRETİM BİRİNCİ SINIF ÖĞRENCİLERİNE İLKOKUMA YAZMA ÖĞRETİMİ İÇİN BİR YAZILIM GELİŞTİRME

Nursel YALÇIN

DOKTORA TEZİ
ENDÜSTRİYEL TEKNOLOJİ EĞİTİMİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ŞUBAT 2006
ANKARA

KABUL VE ONAY SAYFASI

Nursel YALÇIN tarafından hazırlanan KONUŞMA TANIMA TEKNOLOJİSİ YARDIMIYLA İLKÖĞRETİM BİRİNCİ SINIF ÖĞRENCİLERİNE İLKOKUMA YAZMA ÖĞRETİMİ İÇİN BİR YAZILIM GELİŞTİRME adlı bu tezin DOKTORA tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Ömer Faruk BAY
Tez Yöneticisi

Bu çalışma, jürimiz tarafından Endüstriyel Teknoloji Eğitimi Anabilim Dalında doktora tezi olarak kabul edilmiştir.

Başkan : Prof.Dr. H. Güçlü YAVUZCAN

Üye : Prof.Dr. Ömer Faruk BAY

Üye : Doç.Dr. Muammer GÖKBULUT

Üye : Yrd.Doç.Dr. Ahmet ŞENEL

Üye : Yrd.Doç.Dr.Nursal ARICI

Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

KONUŞMA TANIMA TEKNOLOJİSİ YARDIMIYLA İLKÖĞRETİM BİRİNCİ SINIF ÖĞRENCİLERİNE İLKOKUMA YAZMA ÖĞRETİMİ İÇİN BİR YAZILIM GELİŞTİRME

(Doktora Tezi)

Nursel YALÇIN

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Şubat 2006

ÖZET

Bu çalışmada ilkokuma yazma dönemindeki ilköğretim birinci sınıf öğrencileri için bir yazılım geliştirilmiştir. Geliştirilen yazılım (Konuşma Tanıyan Metin Editörü-KTM) bilgisayara bağlı bir mikrofon aracılığıyla öğretmenin veya öğrencinin konuşmasını tanıyarak ekrana yazmaktadır ve sesli komut bazında programı kontrol eden özelliklere sahiptir. KTM, aynı zamanda konuşmacıdan bağımsız ve sürekli konuşma tanıma yapabilen bir yazılımdır. KTM' de konuşma tanıma teknolojisi kullanılmış olup, öğretmen bu yazılımla öğrencilere ilkokuma ve yazma öğretiminde bulunabilecektir. KTM sayesinde ilkokuma ve yazma döneminde hem öğrenci hem de öğretmen açısından etkili bir süreç sağlanabilecektir. KTM' de 30' a yakın sesli komutla yazılım kontrol edilebilirken, konuşulan 10.000 kadar cümlenin veya kelimenin tanınması da söz konusudur. Yazılımın çalıştırıldığı ortama göre bu sayı arttırılabilmektedir.

Bilim Kodu : **705.1.014**
Anahtar Kelimeler : **Konuşma tanıma, Saklı Markov Modelleri, Konuşmacıdan bağımsızlık, İlkokuma - Yazma yöntemleri**
Sayfa Adedi : **127**
Tez Yöneticisi : **Prof. Dr. Ömer Faruk BAY**

DEVELOPING A SOFTWARE FOR TEACHING INITIAL READING-WRITING TO FIRST CLASS STUDENTS OF PRIMARY EDUCATION USING SPEECH RECOGNITION TECHNOLOGY

(Ph.D. Thesis)

Nursel YALÇIN

GAZI UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

February 2006

ABSTRACT

In this study, a software has been developed for first class students of primary education at initial reading-writing term. The software that has been developed (Speech Recognition Text Editor-KTM) recognizes speeches of students or teachers and writes to the screen by microphone that connected to the computer and it has properties that control to the program at voiced command base. Besides, KTM is a software which has independent working from speaker and doing continuous speech recognition. Speech recognition technology was used in KTM, so the teacher can teach first reading-writing to the students. According to the student and also teacher, the effective process can be provided at first reading-writing term by KTM. While the software can be controlled with nearly 30 voiced command by KTM, it is possible to recognition about 10 000 sentence or statement. According to the running area, these numbers can be increased.

Science Code : **705.1.014**
Key Words : **Speech recognition, Hidden Markov Models, Speaker independent, Initial Reading - Writing methods**
Page Number : **127**
Adviser : **Prof. Dr. Ömer Faruk BAY**

TEŞEKKÜR

Tez çalışmam boyunca benden desteğini esirgemeyen Sayın Onur ÇİLİNGİR' e, ses kayıtlarını oluşturmamda bana yardımcı olan tüm öğrencilerime, yeğenlerim Seher YALÇIN, Hasan YALÇIN, Ahmet Evren KUŞ ve Huriye KUŞ' a, kardeşim Alemdar YALÇIN' a ve annem Şükrüye YALÇIN' a, tez danışmanlığımı yapmasından büyük onur duyduğum danışman hocam Sayın Prof. Dr. Ömer Faruk BAY' a bana verdikleri desteklerinden dolayı çok teşekkür eder şükranlarımı sunarım.

İÇİNDEKİLER

Sayfa

ÖZET iii

ABSTRACT iv

TEŞEKKÜR ..... *v*

İÇİNDEKİLER vi

ÇİZELGELERİN LİSTESİ x

ŞEKİLLERİN LİSTESİ xi

SİMGELER VE KISALTMALAR xiv

1. GİRİŞ 1

2. KONUŞMA TANIMA TEORİSİ, HTK YAZILIM MİMARİSİ VE ALGORİTMALARIN TANIMI 10

2.1. Konuşma Tanıma Sistemlerinin Sınıflandırılması 13

2.1.1. Ayrık sözcük tanıma 13

2.1.2. Sözcük yakalama sistemleri 14

2.1.3. Sürekli konuşma tanıma 15

2.2. Konuşma Tanımada Kullanılan Teknikler 23

2.2.1.Yapay sinir ağları 23

2.2.2. Time warping 26

2.2.3. Saklı markov modelleri 27

2.3. HTK Araç Kitinin Özeti 34

2.3.1. HTK yazılım mimarisi 35

2.3.2. HTK aracının genel özellikleri 37

Sayfa

2.3.3. Araç kiti 38
2.4. HTK İle Yapılan Çalışmalar 44
2.4.1. Ses verileri 44
2.4.2. Öznitelik vektörlerinin eldesi 48
2.4.3. Teklises (monophone) eğitimi 48
2.4.4. Üçlüseslere (triphone) geçiş 49
2.4.5. Karar ağacı birleştirimi (decision tree clustering) 50
2.4.6. Karışım sayısının artırılması 51
2.4.7. Çalışma ortamı ve elde edilen dosyalar 51
2.5. Konuşma Tanıma Çekirdeği ve Algoritmaların Tanıtımı 52
2.5.1. Viterbi algoritması 52
2.5.2. Kodların tanıtımı 58
2.5.3. Yöntemler (sınıf metodları – member functions) 59
2.5.4. Çalışma – kullanılma şekli 62
2.5.5. Sınıf metodları 63
2.5.6. Akustik modellerle ilgili sınıflar 68
3. KTM YAZILIMININ GELİŞTİRİLMESİ VE KULLANIMI 72
3.1. Konuşma Tanıma Entegrasyonu 73
3.2. KTM' den XCekirdek' e 74
3.3. XCekirdek' ten KTM' ye 74
3.3.1.Thread yapısı 75

Sayfa

3.4. Yazılım Modüllerinin Tanıtımı 75

3.4.1. Editör penceresi.....76

3.4.2. Ana pencere 76

3.5. Konuşma Tanıyan Metin Editörü – KTM.....77

3.5.1. Metin editörü.....77

3.5.2. Konuşma tanıma 77

3.6. Çalışma Planı 78

3.7. KTM v(1.1) nin Konuşma Tanıma Özelliği 79

3.7.1. Mikrofon ayarları.....79

3.7.2. Fiş tanıma.....80

3.7.3. Komut tanıma 81

4. İLKOKUMA YAZMA ÖĞRETİMİ YÖNTEMLERİ VE KTM' YE UYGULANMASI.....86

4.1. Bireşim (Sentez-Harf-Alfabe) Yöntemi.....86

4.2. Çözümle (analiz-cümle) yöntemi 86

4.3. Karma (karışık) yöntem 86

4.4. Öykü Yöntemi.....86

4.5. Ses Temelli Cümle Yönteminin Aşamaları (yeni sistem) 87

4.6. İlkokuma Yazma Öğretiminin Basamakları.....88

4.6.1. Hazırlık dönemi.....89

4.6.2. Cümle verme dönemi 90

4.6.3. Cümle çözümleme (kelime tanıma) dönemi.....90

Sayfa

4.6.4. Kelime çözümleme (hece tanıma) dönemi.....92

4.6.5. Harf tanıma dönemi.....93

4.6.6. Serbest okuma yazma dönemi.....93

5. SONUÇ VE ÖNERİLER.....107

KAYNAKLAR.....116

EKLER.....119
EK–1 Ses veri tabanında kullanılan bazı cümleler.....120
EK–2 Programı kontrol eden komut listesi.....125

ÖZGEÇMİŞ.....127

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 2.1. Sesli ifade tanıma sistemlerinin sınıflandırılması 16

Çizelge 2.2. Türkçe'nin belli başlı fonemleri.....19

Çizelge 2.3. Saklı markov örneği 31

Çizelge 4.1. Üç farklı öğrenci grubunun oluşturdukları metin.....95

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 2.1. Ses teknolojisi alanları....10

Şekil 2.2. Genel bir konuşma tanıma modeli....11

Şekil 2.3. Sözcük tanıma yapısı....17

Şekil 2.4. Genel bir yapay sinir ağı modeli....24

Şekil 2.5. Basit bir soldan sağa SMM....29

Şekil 2.6. Basit bir soldan sağa SMM tanımı....32

Şekil 2.7. HTK yazılım mimarisi....36

Şekil 2.8. HTK işleme durumları38

Şekil 2.9. Sözcük-altı SMM' lerin eğitimi....41

Şekil 2.10. Bilgisayar sisteminde mikrofon ayarlarının yapılması....45

Şekil 2.11. Kasis 4.0 programında ses kaydetme işleminin başlatılma Aşaması....46

Şekil 2.12. "Şu müzisyenler olağanüstü bir şekilde birbirine uyum sağlıyorlar" cümlesi....47

Şekil 2.13. "Şu müzisyenler olağanüstü bir şekilde birbirine uyum sağlıyorlar" cümlesinin spectrogramı, formant sıklığı ve enerjisi....47

Şekil 2.14. SMM' leri ekleyerek oluşturulan model....53

Şekil 2.15. a)Bütünleşik SMM üzerinden tanıma
b) Sözcüklerin ortak harflerinden faydalanma....54

Şekil 3.1. KTM uygulaması kullanıcı arayüzü tasarımı....72

Şekil 3.2. KTM metin editörü....73

Şekil **Sayfa**

Şekil 3.3. Windows ortamında mikrofon ayarlarının yapılması.....79

Şekil 3.4. Fiş Tanıma ikonuyla ekrana yazdırmak istediğimiz cümleyi seslendirme aşaması 80

Şekil 3.5. Fiş Listesine herhangi bir cümle ekleme veya değiştirme işlemi Aşaması 81

Şekil 3.6. Komut Tanı ikonuna mikrofonla “Tümünü Seç” komutunun Verilmesi 82

Şekil 3.7. Komut Tanı ikonuna mikrofonla “Programdan Çık” komutunun Verilmesi 83

Şekil 3.8. Komut Tanı ikonuna mikrofonla “Yeni” komutunun verilmesi 84

Şekil 3.9. Fiş Tanı ikonuna mikrofonla yeni belgeye cümleler okuma komut tanı ikonuna mikrofonla “Dikey sırala” komutunun verilmesi.....84

Şekil 3.10. KTM metin editörü hakkında bilgi edinmek için Komut Tanı ikonuna mikrofonla “Yardım” komutunun verilmesi 85

Şekil 4.1. Hikayedeki cümleleri fiş listesine ekleme 96

Şekil 4.2. Hikayedeki cümleleri mikrofonla seslendirerek ekrana Yazdırma 97

Şekil 4.3. “Ali ata bak” cümlesinin hikaye içinden “metin ara” komutuyla bulunması.....98

Şekil 4.4. “t” harfinin mikrofona “metin ara” komutu seslendirilerek Bulunması.....100

Şekil 4.5. Öğretilen cümle ve cümleleri küçük harften büyük harfe dönüştürmek için “komut tanı” ikonu kullanılarak mikrofonla “büyük küçük harf” komutu uygulaması.....103

Şekil 4.6. Küçük harfle yazılmış cümlelerin sesli komutla büyük harfe dönüştürme 104

Şekil 4.7. Eğik yazı biçiminin programdaki uygulaması.....105

Şekil 4.8. Hecelerin programın fiş listesine eklenmesi 106

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
KTM	Konuşma Tanıyan Metin Editörü
AST	Ayrık Sözcük Tanıma
SKT	Sürekli Konuşma Tanıma
BST	Bağlı Sözcük Tanıma
SMM	Saklı Markov Modelleri
YSA	Yapay Sinir Ağları
NLP	Natural Language Processing
DTW	Dinamic Time Warping
HTK	Hidden Markov Toolkit
HMM	Hidden Markov Models
DÖK	Doğrusal Öngörü Kodlaması
MLF	Master Label File
MDI	Multiple Document Interface

1. GİRİŞ

Bu tezde amaç, konuşma tanıma teknolojisi yardımıyla ilköğretim birinci sınıf öğrencilerine ilkokuma yazma öğretimini gerçekleştiren bir yazılım hazırlamaktır.

Şimdiye kadar yapılan çalışmalarda ilköğretimde kullanılan konuşma tanıma metoduyla hazırlanmış ilkokuma yazmaya yönelik her hangi bir materyale rastlanamamıştır. Yapılan bu çalışmayla bu eksiklik giderilmeye çalışılmaktadır.

Öğrenci ve öğretmen açısından son derece önemli olan ilkokuma ve yazma dönemine farklı bir yaklaşım olarak daha kolay bir öğretme ve öğrenme imkanı sağlamak için KTM (Konuşma Tanıyan Metin editörü) yazılımı geliştirilmektedir. Öğretmen bu yazılımı uygulayarak öğrencilere ilkokuma ve yazma öğretiminde bulunabilecektir. KTM sayesinde ilkokuma ve yazma adına hem öğrenci hem de öğretmen açısından etkili bir süreç kazanılmış olacaktır. Konuşma tanıma, insan sesinin bir mikrofon kullanarak bilgisayar aracılığıyla tanınmasıdır. Konuşma tanıma teknolojisiyle, kullanıcının (öğretmenin veya öğrencinin) bilgisayara bağlı bir mikrofon aracılığı ile söylediği kelime veya cümlelerin ekrana yazılması sağlanmaktadır.

Günümüzde konuşma tanıma (speech recognition) gerek bu konuda yapılan çalışmaların azlığı gerekse bu alandaki yapılacak çalışmalara gereksinimden dolayı oldukça ilgi çekicidir. Yaklaşık kırk yılı aşkındır yapılan çalışmalar Otomatik Konuşma Tanıma (automatic speech recognition) konusunda insanları oldukça umutlandırmıştır. Gerçekten de konuşarak bilgisayara verilen komutların bilgisayar tarafından anlaşılması ve uygulanması, bir çok yeni ufuğun oluşmasına neden olacak bir teknolojik devrim olarak görülmektedir.

Ticari açıdan bakılacak olursa, konuşma tanıma potansiyel olarak çok büyük pazarı olan bir teknolojidir. Konuşma işleme teknolojisindeki gelişmeler

sayesinde, ağır işitenler daha iyi işitebilmekte, sağır olanlar ise konuşmanın anında yazıya çevrilmesi ile canlı yayınlardaki konuşmaları anlayabilmektedirler (29).

İnsanların biyolojik yapısının tıpa tıp aynı olmayışı çıkardıkları sesin de farklı olmasına yol açmaktadır. Sesin parmak izi gibi güvenilir ayırt edici bir unsur olduğu düşünülerek çeşitli çalışmalar yapılmış ve bunun sonucu olarak konuşmacı tanıma yöntemleri geliştirilmiştir. Örneğin, sadece belirli kişilerin seslerine cevap veren cihazlar, yani güvenliğin gerekli olduğu binalardaki kapılar veya güvenlik kuruluşlarında ses kaydından konuşan kişinin kimliğinin belirlenmesi gibi.

Bu uygulamalarda, insanın değişik durumlarda farklı tonda ve vurguda konuşmaları, duygusal koşulların değişmesinin sese yansıması sonucu çıkarılan seslerin farklı olması ve rahatsızlık gibi bazı olumsuz durumlarda, sesin değişmesi dikkate alındığında kullanılan yöntemlerin çok karmaşık olduğu kestirilebilir. Bu karmaşıklık veri işlemcinin sonuç vermesini geciktirir (29).

Konuşmacının sözcükler arasında 100 - 250 ms' lik duraksamalarla yaptığı konuşmayı tanımak için Ayrık Sözcük Tanıma – AST (Isolated Word Recognition) sistemleri kullanılırken, hiçbir kısıtlama olmaksızın doğal ve 150 - 250 sözcük / dakika kadar hızlı konuşmak için Sürekli Konuşma Tanıma - SKT (Continuous Speech Recognition) sistemleri kullanılır. Bir diğer şekli de sözcükler arasında durmaksızın her sözcüğün açık ve anlaşılır biçimde bastırılarak söylendiği konuşmayı tanıyan Bağlı Sözcük Tanıma - BST (Connected Word Reconition) sistemleridir (18).

Otomatik Konuşma Tanıma Sistemleri, Konuşmacıya Bağımlı (Speaker Dependent) ve Konuşmacıdan Bağımsız (Speaker Independent) olarak da iki ayrı grupta değerlendirilir.

Günümüze kadar yapılmış ve halen yapılmakta olan otomatik konuşma

tanıma ile ilgili çalışmalar, daha ziyade ingilizce dil özelliklerine göre yapılmıştır. Örneğin Dragon Naturally Speaking Preferred ve IBM Via Voice Pro gibi konuşma tanıma programları söz konusudur (6). Bu programın İngilizce, Almanca, Fransızca versiyonları da bulunmaktadır. Ses tanıma ve doğal dil işleme, Microsoft'un hesaplarına göre DOS' tan Windows'a geçişten sonraki en büyük atılım olacaktır. Onlara göre bu teknoloji normalde cansızmış gibi görünen bilgisayarla olan ilişkileri köklü olarak değiştirecektir. Ancak Microsoft' a göre bu teknoloji birden ortaya çıkabilecek bir teknoloji değildir. Sonraki on yıl içinde ortak çalışmalar sonucunda yavaş-yavaş gelişip yerine oturacaktır. Bu düşünceyle hareket eden Microsoft, Office XP programlarında sesli komutlarla çalışan bir araç çubuğu oluşturma imkanı sunmaktadır. Ancak bu imkan sadece İngilizce (ABD), Japonca ve basit Çince dillerinde hazırlanmış sürümlerinde bulunmaktadır. Konuşma tanıma alanında günümüzde yaşanan en büyük zorluklardan biri hiç kuşkusuz, konuşmacıdan bağımsız konuşma tanıma sistemleri için hala yüksek boyutlu sözcüklerde yüksek doğrulukta tanıma oranlarının elde edilememesidir. Özellikle yurt dışında İngilizce veya Japonca dilleri için bu alanda yapılan çalışmaların sayısı tatminkar sonuçlar elde edilemese de oldukça fazladır. Ancak ülkemizde Türkçe için yapılan çalışmalar diğerleri ile karşılaştırıldığında oldukça düşük sayıdadır. Türkçe dil yapısına yönelik yapılmış çalışmalara henüz sıklıkla rastlanamamaktadır. Konuşma tanıma teknolojisiyle Türkiye' de ticari manada ciddi olarak ilgilenen pek fazla firma yoktur. Bu konuda daha çok çeşitli üniversitelerde doktora tezleri şeklinde çalışmalar yapılmaktadır. Ancak bu çalışmalardan çok azı dışardan mali destek alabilmektedir. Bazı yüksek lisans ve doktora çalışmalarına örnek verilecek olursa: (15) "PC Ortamında Sesli Komutları Tanıma" adlı yüksek lisans tezinde Windows 95 / 98 işletim sistemi kullanan kişisel bilgisayarlarda sesli komut yürüten yazılım geliştirilmiştir. Bu yazılım örüntü tanıma yöntemi kullanılarak hazırlanmıştır. Aynı zamanda bu yazılım konuşmacıya bağımlı olarak sınırlı sayıda kelimeyi tanımayı gerçekleştirmektedir. Diğer taraftan (7) "Bir Türkçe Fonem Kümeleme Sistemi Tasarımı ve Gerçekleştirimi" adlı doktora çalışmasında, Türkçe sesli ifade phonlarından fonemlere geçiş çalışmaları yürütülerek sözcük tanıma ve tanınan sözcüğün yazıya

geçirilmesi kapsamında dolaysız yararlanılacak fonem kümeleme amaçlanmıştır. Bu çalışmada tanıma sürecinde kullanılacak ve her fonemi en iyi temsil ettiği varsayılan özellik vektörlerinden Türkçe Codebook oluşturulmuştur. Diğer bir çalışma olarak (10) "Yapay Sinir Ağları İle Kişilerin Ses Örneklerinden Kimliklerinin Tanınması" adlı yüksek lisans tezinde, YSA yöntemi olarak 'hatanın geriye yayılarak düzeltilmesi' (Backpropagation) algoritması ile çıkışta '0' ve '1' çıktısı verecek şekilde eğitilerek, konuşan kişinin tanınması uygulaması yapılmıştır. Bir başka çalışma olan (18) "Yapay Sinir Ağları ile Ayrık Türkçe Sözcüklerin Tanınması" adlı yüksek lisans tezinde ise çok katmanlı ileri beslemeli ağlar kullanılarak Türkçe ayrık sözcüklerin tanınmasına çalışılmıştır. (26) "Bir Türkçe Sesli İfade Tanıma Sisteminin Kural Tabanlı Tasarımı ve Gerçekleştirimi" adlı yüksek lisans tezinde, fonem tabanlı olarak konuşmacıya bağımlı bir ayrık sözcük tanıma sistemi üzerine çalışılmıştır.

Bu konuyla ilgili olarak GVZ yazılım ses teknolojileri firmasının yapmış olduğu çalışmalar vardır. Bu grubun yapmış olduğu çalışma sadece metni konuşmaya dönüştürme (text to speech) şeklinde olup Türkçe yazılan metni tam bir insan sesiyle seslendirmektedir ve ancak konuşmayı metne dönüştürme (speech to text) teknolojisiyle ilgili çalışmalarına da devam etmektedir. Diğer taraftan TÜBİTAK ODTÜ BİLTEN' de bulunan proje grupları arasında Konuşma İşlemeye yönelik çalışmalar söz konusudur. Ayrıca Bilkent Üniversitesi, Türkçe Dil Ve Konuşma İşleme Merkezi kurarak bu konuyla ilgili araştırmalarını sürdürmektedirler.

Önemli olan bağlı konuşmayı (connected speech) üretmek ve tanımaktır. Küçük bir sözcük dağarcığından bir veya iki kelimelik komutlar seçilerek basit tanıma işlerinden, binlerce farklı kelime içeren tam cümlelere yani, bağlı konuşmaya ulaşmaya çalışmak, var olan tekniklerin çoğalmasını veya hesaplama gücünün artmasını gerektirdiğinden basit değildir. Yeterli performansı doğru olarak elde etmek için konuşma ve dil hakkındaki bütün yeni bilgi türlerinin muhtemel konuşma algoritmalarında yorumlanması gerekecektir (18).

Otomatik konuşma tanıma konusunda söz edilmesi gereken en önemli nokta, kullanılan yöntem ve algoritmalardır. Şu ana kadar bir çok değişik yöntem ve her yönteme ait çok sayıda algoritma önerilmiştir. Burada amaç hiç kuşkusuz bir yandan konuşma tanıma sırasındaki yanılma payını en azda tutmaya çalışırken, diğer yandan da işlem yükünü ve dolayısıyla zaman gecikmesini düşürmektir. Tabii bunlara, tanınacak kelime sayısının mümkün olduğunca çok olması, diğer bir deyişle ayırt edilebilirliğin yüksek olması yapının genişlemeye ve değiştirilebilmeye uygun olması, yani esneklik gibi kısıtlayıcı etkenlerde eklenebilir. Şu ana kadar yapılan bir çok otomatik konuşma tanıma uygulaması çoğunlukla Saklı Markov Modeli, SMM (Hidden Markov Models) adı verilen yöntemle, bunun üzerine oturtulmuş algoritmalarla gerçekleştirilmiştir.

Diğer yandan son yıllarda sıklıkla kullanılan bir yöntem olan Artificial Neural Networks (Yapay Sinir Ağları, YSA) kullanılmasıdır. Yapay zekanın alanına giren ses uygulamaları içinde ses tanıma ile beraber aynı zamanda konuşmacı tanımlama ve ses üretmede vardır. Bu üç başlık konuşma işleme (speech processing) veya ses işleme başlığı altında toplanır. Ses tanıma bu üç konudan en zor olanıdır. Yani konuşma tanıma uygulamalarının konuşmacıdan bağımsız olması için ayrıca çaba harcanmaktadır. Ses üretme (speech synthesis) sistemiyle ilgili kısmen de olsa üzerinde çalışılmış belli başlı bazı algoritmalar vardır. Kabre ve Spalanzi' ye göre hem genetik algoritma hem de yapay sinir ağları birlikte kullanılarak konuşma tanıma ve kavrama problemlerine çözüm getirilebilmektedir (21). Onlar EVERA adı verdikleri bir model ile ses işleme uyumlu bir evrimsel programlama çevresi oluşturmuşlardır. (5)' ya göre ise konuşma tanıma işlemleri ne tek Hidden Markov Model'e göre ne de tek YSA' yla yapılmamalıdır. Albesano' ya göre hem SMM' kullanılarak hem de YSA kullanılarak konuşma tanıma işlemi yapılabilir. Böylelikle Hybrid SMM - YA olarak yeni bir melez model ortaya koymuşlardır.

Konuşma tanıma için üç ana işlem gerekir. İlki bir ön işlemciyle konuşma dalga şeklinden önemli bilgiyi çıkarmaktır. Pek çok tanıyıcı da konuşma

taslağından spektral parametreleri içeren giriş örüntüsü, Hızlı Fourier Dönüşümü (Fast Fourier Transform) veya Lineer Tahmin Kodlaması (Linear Predictive Coding) teknikleri kullanılarak her 10 ms' de alınır. İkinci olarak ön işlemciden alınan giriş örüntüsü, yerel taslak uzaklıklarını hesaplamak için kelime modellerinde saklanmış örneksel örüntülerle karşılaştırılır. Üçüncü adımda ise yerel uzaklıklar, kelime modellerini oluşturan saklanmış örneksel örüntü dizilerine zaman düzenlemesi yapmak ve tüm kelime karşılaştırma değerlerine ulaşmak için kullanılır. Bir otomatik konuşma tanıma sisteminin iki aşaması vardır. Biri eğitim aşaması diğeri de test aşamasıdır.

Genelde konuşma tanımada kullanılan alanların bazıları aşağıda sıralanmıştır:

- İşitme engelliler, insan-bilgisayar komut iletişiminde,
- Ses ile yazı yazma,
- Hava alanlarında,
- Robotlar,
- Mekanik araçları uzaktan yönetmek,
- Zeki telefon sistemleri,

Büyük bir ticari önem taşıdığından birçok üniversite ve bazı firmalar kendi bünyelerinde konuşma tanıma konusunda araştırmalarına devam etmektedirler. Tanıma işleminin yapay olarak bilgisayarda uygulanması için ilk adım olarak konuşma tanımada karşılaşılan problemlerin tanımlanması gerekir. Daha öncede belirtildiği gibi konuşma tanıma konusu, genel olarak bir çok alanı ilgilendirmektedir. Bunlar matematik, istatistik, psikoloji vb gibi alanlardır.

Tutarlı ve mükemmel güvenilir konuşma tanıma bir çok sebepten dolayı zordur. Mesela sorunlardan birisi sesin insandan insana değişmesidir. Çoğu kişi seslerinde öksürme, hapşurma, duraklama gibi rahatsız edici gürültüler bulundurur ve herkes aynı kelimeyi her zaman aynı nitelikte telaffuz etmez.

Diğer bir tanesi ise, bazen sesi kirleten sinyaller barındırır. Bu ise sesin farklı algılanmasına ve anlaşılmamasına sebep olur. Telefon hatlarında konuşma tanıma olayını sağlamak için konuşma tanıma programları hat gürültüsünün değişken dereceleri için değişik gürültüyü yok eden sistemler geliştirmişlerdir. Diğer bir sorun ise birçoğumuzun konuşma tarzıdır. Konuştuğumuz zaman genelde yanlış telaffuz edilen veya gereksiz kullanılan (um, ee, hımm) kelimeleridir. Kelimeler aynı zamanda arka arkaya söylendiğinde yutulabilirler. Buna ilaveten, bir çok kelime birbirine benzer. Maddeler halinde bu sorunlar sıralanırsa (15);

- Akıcı bir konuşmada kelimeler arasında boşluklar olmayabilir. Bu konuşmayı tanımaya çalışan yazılım için büyük sorunlar çıkarır,
- Sesler kişiden kişiye çok değişik olabilir. Aynı kişi bile aynı kelimeyi değişik bir şekilde telaffuz edebilir,
- Kadın, erkek ve çocuk sesleri arasında büyük farklılıklar vardır,
- Konuşmanın yapıldığı ortamda tanımayı zorlaştıran başka seslerde bulunabilir. Bu seslerin ana sesten ayrıştırılması işlemi zordur,
- Konuşmacının sesi tam olarak anlaşılmasa bile insan gelen kelimeyi konuyu bilmesinden, kelimenin gelişinden dolayı anlayabilir. Böyle bir tanımayı bilgisayara uygulamak zordur.

Normal işitsel konuşmayı yapay görsel konuşmaya güvenilir olarak çeviren sistem, aşağıdaki unsurları gerektirmektedir (25):

- İşitsel - görsel konuşmanın tanımlanmış bir veri tabanı,
- Hem görsel hem işitsel konuşmanın tasviri,
- Her iki tasvir arasındaki ilişkiyi tanıtan metodlar ve
- Verilen işitsel konuşmayı görsel konuşmayla sentezleyen teknik.

İşitsel konuşmanın bir çok nitelendirilmiş veri tabanı vardır. Ama görsel konuşmanın veri tabanı kolaylıkla elde edilemez (25).

İnsanlığın sahip olduğu çağdaş uygarlığın büyük ölçüde okuma yazmaya dayanan bir uygarlık olduğu söylenebilir. Bu kadar büyük bir işleve sahip olan okuma yazma becerisinin temeli ilkokuma - yazma çalışmaları ile atılmaktadır. İlkokuma - yazma öğretiminin genel amacı, çocuğa yaşamı boyunca kullanacağı okuma- yazma temel becerilerini kazandırmaktır.

İlkokuma - yazma etkinlikleri ve kullanılan yöntemler, toplum olarak eğitim felsefemizin, eğitime bakışımızın, öğrenciye yansıtıldığı ilk uygulamalardır.

İlkokuma - yazma öğretimi, harflerin görülüp tanınması son aşamasından sonra, mekanik olarak onları seslendirmeyi içeren süreçten öte bir şeydir. Bu süreç, harf, kelime, cümleyi tanıma ve bir bütün olarak metni anlamlandırmayı içeren karmaşık bir süreçtir.

Süreci karmaşık hale getiren anlama ve anlamlandırma sürecindeki işlemlerin her öğrencide farklı olmasıdır. İlkokuma ve yazma dönemine farklı bir yaklaşım olarak daha kolay bir öğretme ve öğrenme imkanı sağlamak için konuşmacıdan bağımsız çalışan ve sürekli konuşma tanıyan bir özelliğe sahip olacak şekilde bir yazılım geliştirilmektedir. Sesli komutlarla da yazılımı kontrol etmek mümkündür. Konuşma tanıma sistemi saklı markov modelleri kullanılarak oluşturulmaktadır.

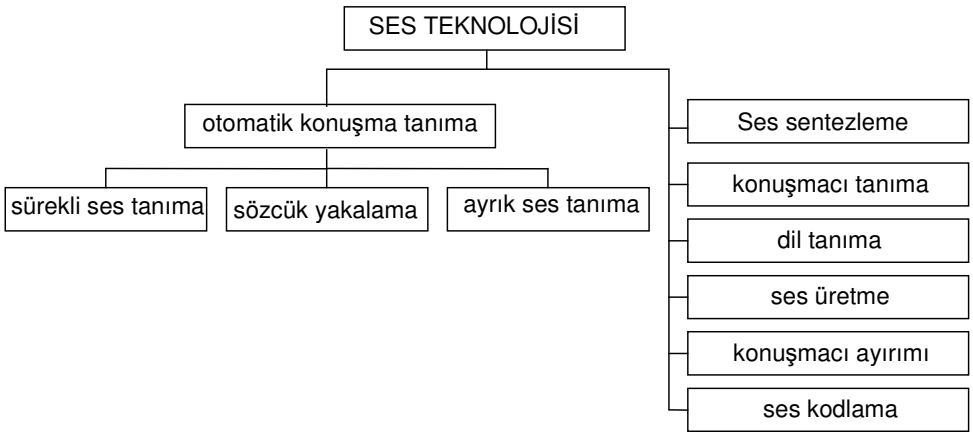
Bu çalışmada, insan sesinin bir bilgisayar aracılığıyla tanınması sağlanarak, ilköğretim 1. sınıf öğrencilerine ilkokuma yazma öğretimine yönelik bir yazılım hazırlanmaktadır. Bu tanınma için öncelikle sesin dinlenmesi işlemi gerçekleştirilmektedir. Dinleme işleme işlemi gerçekleştirilirken, ses normal bir mikrofon ve ses kartı aracılığıyla örneklenerek alınmaktadır.

Tezin 1. Bölümünde tezin içeriğine yönelik giriş yapılarak çalışmanın amacı vurgulanmış ve konuşma tanıma çalışmaları hakkında bilgi verilmiştir. Tezin 2. Bölümünde konuşma tanıma teknolojisi genel hatlarıyla anlatılmıştır. Ayrıca ses modellerinin eğitimi için kullanılan HTK sistemi de çok genel

olarak açıklanmıştır. Yine bu bölümde HTK ile yapılan çalışmalar ve aynı zamanda oluşturulan yazılıma ait kodlar tanıtılmaktadır. Tezin 3. Bölümünde ise oluşturulan konuşma tanıma sistemi tasarımı ve KTM yazılımının modülleri anlatılmakta ve kullanımı açıklanmaktadır. Tezin 4. Bölümünde ise hazırlanan konuşma tanıma editörünün ilköğretim birinci sınıf öğrencileri için ilkokuma - yazma öğretimi yöntemleriyle kullanımı anlatılmaktadır. Tezin 5. Bölümünde ise hazırlanan çalışmaya dair elde edilen sonuçlar dile getirilmiştir.

2. KONUŞMA TANIMA TEORİSİ, HTK YAZILIM MİMARİSİ ve ALGORİTMALARIN TANIMI

Ses teknolojisi yedi ortak konuşma uygulamalarını içermektedir. Şekil 2.1 de ses teknolojilerinin çeşitleri görülmektedir.



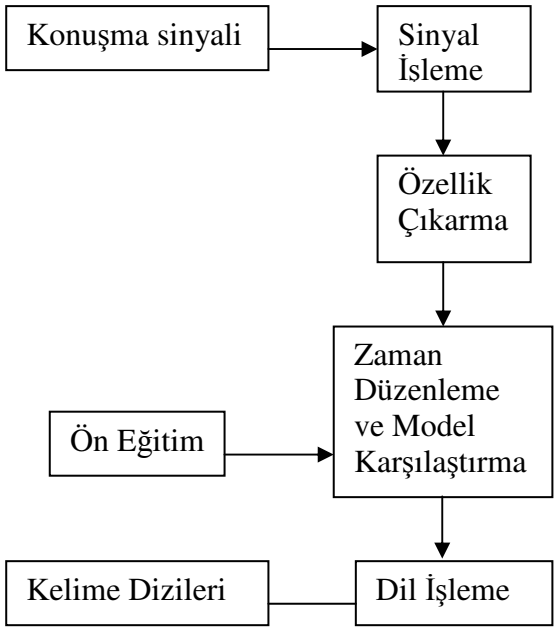
Şekil 2.1. Ses teknolojisi alanları

Ses teknolojisi içerisinde bulunan otomatik konuşma tanıma, sürekli konuşma tanıma, ayrık sözcük tanıma ve kelime yakalama sistemlerini içerisinde bulundurmaktadır. Diğer taraftan ses sentezleme, konuşmacı tanıma, dil tanıma, ses üretme, konuşmacı ayırımı ve ses kodlamayı da içine alan ses teknolojileri içinde en zor olan alan otomatik konuşma tanımayla ilgili olan alandır (27).

Otomatik Konuşma tanıma problemi, verilen bir akustik X dizisi için, W kelime dizilerini bulmak için oluşturulmuştur. Konuşma cümleleri, $W = (w_1, w_2,\ldots,w_t)$ şeklinde belirtilen kelimelerin dizisi olarak gösterilir. w_t, ayrık bir t zamanında söylenmiş belli bir kelimedir. Kelimelerin dizisi söylenen sesli ifade ile bağlantılıdır ve bu sesli ifade X olarak gösterilen akustik sesler dizisidir. Otomatik konuşma tanıma oldukça zor ve karmaşık bir problemdir. Matematiksel olarak bir f fonksiyonu $f: X \rightarrow W$ şeklinde gösterilir (9).

Otomatik konuşma tanıma teorisi oldukça geniş bir konudur. Genel olarak bir örüntü tanıma (pattern recognition) problemi olarak görülebilir. Konuşma sinyalinin stokastik yapısı istatistiksel yöntemlerle konuşma tanıma yapılmasını gerekli hale getirmektedir. Genel bir konuşma tanıma modeli şu modüllerden oluşmaktadır (Şekil 2.2) (27,18).

- Konuşma tanıma sinyalinin bir gösterimini elde etmek için sinyal işleme modülü
- Bu gösterimin anahtar elemanlarını belirlemek ve fazla bilgiyi çıkarmak için özellik çıkarma modülü
- Kelime tespitini yapmak için zaman düzenleme ve model karşılaştırma algoritmaları
- Bir final kelime dizisi seçmek için dil modeli



Şekil 2.2. Genel bir konuşma tanıma modeli (27)

Konuşma sinyalinin modellenmesi, sesli ifadelerin, bilgisayar destekli olarak tanıma sürecine sokulabilmesi için bunların öncelikle bu sürece hazırlanmaları gerekmektedir.

Bu amaçla sesli ifadelerin bir mikrofon aracılığıyla örneksel sinyallere dönüştürülmesi, sayısallaştırılması, sayısallaştırılan bu sinyallerin gerekirse filtrelenmesi, etiketlenmesi (örneğin sesler, fonemler, sözcükler olarak) ve tanıma işlemlerine taban oluşturacak parametrik yapılar ya da yalın modellerle ifade edilen biçimlere dönüştürülmesi gerekmektedir.

Sinyal işleme modülünün amacı, örneklenen konuşma sinyalini işlemek ve genlik değişimlerinden, konuşmacı stresinden, iletişim araçlarından veya kanalından gelen gürültüden bağımsız bir gösterim elde etmektir.

Özellik çıkarım modülü, sinyaldeki geçişleri yakalayan parametrelerin kümesini hesaplar ve her fonemi göstermek için yeterince güçlüdür. Bu parametreler genellikle özellikler olarak adlandırılırlar ve genellikle sabit-zaman aralıklarında hesaplanırlar. Sinyaldeki geçişler, konuşma sinyalindeki fonetik bilginin kodlanmasını gösterebilen önemli ipuçlarıdır.

Zaman düzenleme ve model karşılaştırma modülü, bir kelimeyi, o kelimenin verilen gösterimlerine dayanarak karşılaştırmayı dener. Zaman düzenleme, modellenmiş akustik veya fonetik olayların düzenlenmesi için, konuşma oranındaki değişikliklerden dolayı "sözdeki zaman bozulmalarını" referans alır. Bu zaman bozulması veya geçici değişimler, doğal olarak olur ve özellikle konuşma oranındaki değişimler seslilerin süresini önemli şekilde etkiler. Zaman düzenleme ve model karşılaştırma, bir kelime modelini formüle etmek için kullanılan eğitim yöntemlerine bağlıdır. Kelime modeli, eğitim safhası esnasında çıkarılan parametrelerin kümesinden oluşur.

Dil işleme modülü, konuşma tanıma işleminin son aşamasıdır. Hangi dile yönelik konuşma tanıma sistemi hazırlanacaksa o dile ait çeşitli kuralların bilinmesi ve bu kurallara uyarak çalışılması gerekmektedir. Dile ait bu yapılar bilinerek kelime seçimlerinin oluşturulmasıyla ilgili modüldür.

Konuşma tanıma sistemleri kabul ettikleri girdi ses sinyalinin ve içermesi beklenen metin karşılıklarının yapılarına göre çeşitli dallar altında incelenir:

- Ses sinyalinin tek veya çok kişiye ait olabilmesi: Konuşmacıya bağlılık
- Tanınacak metin kümesinin genişliği: Dağarcık
- Tanınacak seste metin elemanlarının yerleşimi: Sürekli, bağlı veya ayrık tanıma

2.1. Konuşma Tanıma Sistemlerinin Sınıflandırılması

Konuşma tanıma programları, alanların ihtiyacına göre tasarlanır. Bu alanlarda çıkan sorunların çözümü için değişik sistemler ve metotlar geliştirilmiştir. Sesli ifade sistemleri ya da sesli ifade tanıyıcılar, artan zorluk sırasına göre aşağıda sıralanmıştır:

- Ayrık sözcük tanıma sistemleri (isolated word recognition systems),
- Sözcük yakalama sistemleri (word spotting systems),
- Sürekli konuşma tanıma sistemleri (continous speech recognition system).

2.1.1. Ayrık sözcük tanıma

Yalıtılmış konuşma tanıma olayı kelime haznesi uyuşmasının en temel formudur ve bu sistem herhangi bir şey kullanıcıya sorulduğunda kullanıcının tek kelimelik bir girdi yapmasını bekler. Sözcükler arası duraklar olmak zorundadır (20). Tanıma sözcük tanımadır. Konuşma tanımada zor olan noktalardan birisi de sözcüklerin başlangıç ve bitiş noktalarının belirlenmesidir. Ayrık sözcük tanımada sözcüklerin birbirinden bağımsız telaffuz edilmesi tanımayı kolaylaştırır. Bu yöntem sadece belirli kelimelerin tanınmasının yeterli olduğu alanlarda kullanılır. Özetle;

- En yalın konuşma tanıma şeklidir.
- Sözcük yakalama ve SKT' ya temel oluşturur.
- Sözcükler arasındaki duraklar belirgindir ve tanıma kolaylaşmaktadır.
- Sözcükler birbirinden bağımsız olarak ele alınır.

- Dolayısıyla birlikte seslendirme sorunu yoktur.

2.1.2. Sözcük yakalama sistemleri

Sürekli konuşma içinde belirli bir kelimenin ortaya çıkışını belirleme işlemidir. Bu tür uygulamalarda en başarılı sonucu veren dinamik zaman sıkıştırma programlama teknikleri kullanılır. Yakalanacak her kelime şablon tarafından gösterilir. Sadece şablonlar tarafından bilindiğinden dolayı, sesin bitiş noktalarından bağımsız kılmak önemlidir. Potansiyel başlangıç noktası olarak giren ses akıntısının her örneği süreç olarak kabul edilmelidir. Tanıma işlemi, aranan şablonun sesli ifade içinde çakıştığı bir örüntü arama biçiminde gerçekleşmektedir.

Konuşma tanıma sistemleri geliştiricileri insanların normal olarak anlaşılmayan kelimeleri konuşmalarda kullanmaları gerçeği üzerinde durmuşlardır (daha öncede bahsi geçen “um” ”eee” ”hımm” gibi). Alışıla geldik bir konuşmada bu kelimeleri geçersiz olarak algılar ve bunun yerine mesaj taşıyan gerçek kelimeler üzerine yoğunlaşılır. Diğer taraftan bilgisayarlar çok zeki değildirler ve bu ek sesler sorun çıkarabilir.

Araştırmacılar anahtar kelimeleri tanımak için bilgisayarlara insanların doğal olarak ne yaptıklarını öğretirler. Sözcük yakalama olarak adlandırılan bu teknik bilgisayara cümle içinde kelimeyi tanımayı mümkün kılar. Örnek: “ben Kayseri’ye gidiyorum” da olan “Kayseri’ye “ gibi. Özetle;

- Konuşma içinde aranan sözcüklerin yakalanmasını sağlar.
- En çok dinamik zaman eşleştirme tekniğinde kullanılır.
- Her sözcük bir şablon ile ifade edilir.
- Tanıma işlemi aranan şablonun konuşma içinde çakıştığı bir örüntü tanıma şeklinde olur.

2.1.3. Sürekli konuşma tanıma

Sürekli sesli ifade tanıma sistemleri, bağlı sözcük tanıma (Connected Word Recognition) sistemleri ve karşılıklı konuşma tanıma sistemleri (Conversational Speech Recognition) olmak üzere iki gruba ayrılabilir. Bunlardan ilki tanıma işlemini sözcük bazında yapmayı hedeflerken, ikincisi cümlenin anlamının da anlaşılmasını hedefler. Bu sebeple karşılıklı konuşma tanıma sistemleri sesli ifade anlama (speech understanding) sistemleri olarak da adlandırılabilir ve karmaşık dilbilgisi kurallarının da sistemde yer almasını gerektirir. Bu kısım günümüzde ayrı bir alan olarak ele alınmıştır. Bu alan Doğal Dil İşleme ya da Anlama (Natural Language Processing / Understanding, NLP) olarak adlandırılmaktadır. Konuşma tanıma mimarisi içindeki yeri önemlidir.

Sürekli sesli ifade tanımadaki zorluk üç değişik nedenden kaynaklanmaktadır. Bunlardan ilki, sürekli ifade içindeki sözcük sınırlarının belirgin olmamasıdır. Sürekli sözcük tanımada, sözcük sınırlarının bulunması zor, hatta kimi zaman olanaksızdır. İkinci sorun, tanınacak ses birimlerinin, ön ve arkasına gelen diğer ses birimleri tarafından etkilenmeleridir. Üçüncü sorun vurgulama, duraklama gibi bürünlerden kaynaklanmaktadır. İsim, sıfat ve fiiller net ve yüksek enerjili seslerle ifade edilirken aradaki bağlaçlar ve kısa süreli sözcükler çoğu kez düşük enerjili olmakta ya da yutulmaktadır. Örneğin, “bir gün” kelime çifti genellikle “bigün” olarak seslendirilir. Özetle;

- Ayrık sözcük tanıma ve sözcük yakalama sistemlerine göre daha zordur.
- Bu zorluk üç nedenden kaynaklanmaktadır.
- Sözcük sınırları belirgin değildir.
- Tanınacak söz birimlerinin ön ve arkasına gelen ses birimleri tarafından etkilenirler. Seslerin birlikte seslendirilmesinden kaynaklanan bir sorundur.
- Sözcüklerin söyleyişte, hızlı konuşma söz konusu olduğu için, kaybedilmesi söz konusudur.

Sürekli ya da ayrık olmalarının dışında sesli ifade tanıma sistemleri konuşmacıya bağımlılığına göre de ikiye ayrılır (Çizelge 2.1).

- Konuşmacıya bağımlı sesli ifade tanıma sistemleri (Speaker Dependent).
- Konuşmacıdan bağımsız sesli ifade tanıma sistemleri (Speaker Independent).

Bunlardan ilki tek bir kişi için referans şablonlarının oluşturulmasını öngörür. Yeni kişilerin konuşmalarının tanınabilmesi için referans alınan şablonların günlenmesi gerekir. İkincisinde ise sistem herhangi bir kişi tarafından seslendirilen bir sesli ifadeyi tanıyabilir. Doğal olarak bir sistemin kullanılış alanını artırmak için amaç kişiden bağımsız bir sesli ifade tanıma sistemi olmalıdır. Fakat bunu başarmak kişiye bağımlı bir sistem geliştirmekten daha zordur (26).

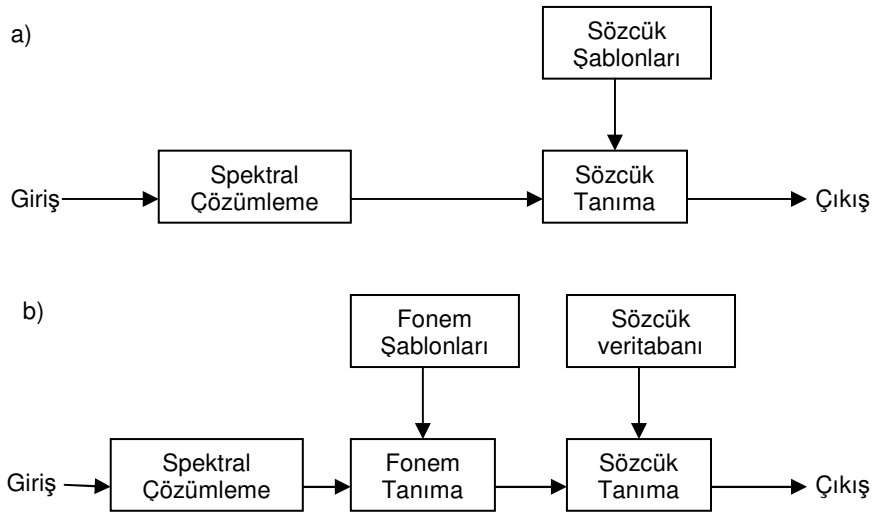
Çizelge 2.1. Sesli ifade tanıma sistemlerinin sınıflandırılması

Sesli ifade tanıma		
1. Sözcük tabanlı ayrık sesli ifade tanıma 2. Sözcük tabanlı sürekli sesli ifade tanıma - Bağlı sözcük tanıma - Karşılıklı konuşma tanıma	1. Kisiye bağımlı sesli ifade tanıma 2. Kişiden bağımsız sesli ifade tanıma	1. Sözcük tabanlı sesli ifade tanıma 2. Fonem tabanlı sesli ifade tanıma

Bir diğer sınıflandırma da yukarıda söz konusu edildiği gibi tanıma için seçilen birimin büyüklüğüne göre yapılabilir. Bunun için de sesli ifade tanıma sistemlerini ikiye ayırmak mümkündür (26).

- Sözcük tabanlı sesli ifade tanıma sistemleri; tanıma için öngörülen en küçük birim olarak sözcüğün kabul edildiği sistemlerdir (word based speech recognition).
- Fonem tabanlı sesli ifade tanıma sistemleri; tanıma için fonemlerin en küçük birim olarak kabul edildiği sistemlerdir (phoneme based speech recognition).

Bu sistemlerden ilkinin doğruluk derecesi daha yüksek olmaktadır. Çünkü fonemler arası geçişlerin olumsuz etkileri burada gözlenmez. Ancak, bunun yanında sürekli sesli ifade tanıma söz konusu olduğunda sözcükler arası geçişler yine sorun olacaktır. Ayrıca sözcük tabanlı sesli ifade tanıma sistemlerinde referans şablonu olarak sözcüğün tamamı alındığından ve bir dilde çok sayıda sözcük olduğundan sistemin gereksinim duyduğu işleyici zamanı ve bellek gereksinimi çok daha fazla olacaktır. Fonem tabanlı tanımada ise doğruluk yüzdesi bir miktar düşerken, çok az olan fonem sayısı, hızlı sonuç üretme olanağı sayesinde, hata azaltma amaçlı geri dönüşleri mümkün hale getirmektedir. Sözcük ve fonem tabanlı sistemlerden başka bu ikisi arasında sözcük altı birimleri temel alan sesli ifade tanıma sistemlerini de söz konusu etmek mümkündür (26).



Şekil 2.3. Sözcük tanıma sisteminin yapısı: (a) Sözcük tabanlı sesli ifade tanıma; (b) Fonem tabanlı sesli ifade tanıma (17)

Şekil 2.3' deki sözcük tanıma sistemlerinin ikisinde de amaç tüm sözcüğün tanınmasıdır. Birinci yaklaşımda, sesli ifadeden doğrudan sözcük tanımaya geçilirken ikinci yaklaşımda önce fonem tanıma işlemi yapılmakta, daha sonra bir sözcük veritabanı kullanılarak bu fonemler sözcüklere dönüştürülmektedir. Daha önce de söz konusu edildiği gibi (a) şeklindeki sistemin doğruluğu daha fazla fakat tanıyabildiği sözcük sayısı az olacaktır. (b) seklindeki sistemde ise sözcük sayısı için başlangıçta bir sınır yoktur. Bir sözcük veritabanına sözcük eklemek doğal olarak o sözcük için sesli ifade şablonunun sisteme öğretilmesinden daha kolay olduğu için (b) sekli genişlemeye daha elverişlidir (26).

Sesli ifade tanımada, sözcük yerine, fonem gibi alt ses birimlerine dayanılabilir. Bir fonem, genellikle, konuşmanın en küçük birimi olarak tanımlanır. Her dilin standart lehçelerinin özgün fonem grupları vardır (2). Ancak bu ses birimlerinin sayıca kısıtlı olması ve sözcükleri bunların kombinasyonu ile ifade edebilme özelliğinin bulunması gerekmektedir (Çizelge 2.2) (16).

Çizelge 2.2. Türkçe' nin belli başlı fonemleri (16)

	A		**E**		**I**		**İ**		**O**		**Ö**		**U**		**Ü**	
B	AB	BA	EB	BE	IB	IB	İB	Bİ	OB	BO	ÖB	BÖ	UB	BU	ÜB	BÜ
C	AC	CA	EC	CE	IC	CI	İC	Cİ	OC	CO	ÖC	CÖ	UC	CU	ÜC	CÜ
Ç	AÇ	ÇA	EÇ	ÇE	IÇ	ÇI	İÇ	Çİ	OÇ	ÇO	ÖÇ	ÇÖ	UÇ	ÇU	ÜÇ	ÇÜ
D	AD	DA	ED	DE	ID	DI	İD	Dİ	OD	DO	ÖD	DÖ	UD	DU	ÜD	DÜ
F	AF	FA	EF	FE	IF	FI	İF	Fİ	OF	FO	ÖF	FÖ	UF	FU	ÜF	FÜ
G	AG	GA	EG	GE	IG	GI	İG	Gİ	OG	GO	ÖG	GÖ	UG	GU	ÜG	GÜ
Ğ	AĞ	ĞA	EĞ	ĞE	IĞ	ĞI	İĞ	Ğİ	OĞ	ĞO	ÖĞ	ĞÖ	UĞ	ĞU	ÜĞ	ĞÜ
H	AH	HA	EH	HE	IH	HI	İH	Hİ	OH	HO	ÖH	HÖ	UH	HU	ÜH	HÜ
J	AJ	JA	EJ	JE	IJ	JI	İJ	Jİ	OJ	JO	ÖJ	JÖ	UJ	JU	ÜJ	JÜ
K	AK	KA	EK	KE	IK	KI	İK	Kİ	OK	KO	ÖK	KÖ	UK	KU	ÜK	KÜ
L	AL	LA	EL	LE	IL	LI	İL	Lİ	OL	LO	ÖL	LÖ	UL	LU	ÜL	LÜ
M	AM	MA	EM	ME	IM	MI	İM	Mİ	OM	MO	ÖM	MÖ	UM	MU	ÜM	MÜ
N	AN	NA	EN	NE	IN	NI	İN	Nİ	ON	NO	ÖN	NÖ	UN	NU	ÜN	NÜ
P	AP	PA	EP	PE	IP	PI	İP	Pİ	OP	PO	ÖP	PÖ	UP	PU	ÜP	PÜ
R	AR	RA	ER	RE	IR	RI	İR	Rİ	OR	RO	ÖR	RÖ	UR	RU	ÜR	RÜ
S	AS	SA	ES	SE	IS	SI	İS	Sİ	OS	SO	ÖS	SÖ	US	SU	ÜS	SÜ
Ş	AŞ	ŞA	EŞ	ŞE	IŞ	ŞI	İŞ	Şİ	OŞ	ŞO	ÖŞ	ŞÖ	UŞ	ŞU	ÜŞ	ŞÜ
T	AT	TA	ET	TE	IT	TI	İT	Tİ	OT	TO	ÖT	TÖ	UT	TU	ÜT	TÜ
V	AV	VA	EV	VE	IV	VI	İV	Vİ	OV	VO	ÖV	VÖ	UV	VU	ÜV	VÜ
Y	AY	YA	EY	YE	IY	YI	İY	Yİ	OY	YO	ÖY	YÖ	UY	YU	ÜY	YÜ
Z	AZ	ZA	EZ	ZE	IZ	ZI	İZ	Zİ	OZ	ZO	ÖZ	ZÖ	UZ	ZU	ÜZ	ZÜ
	/A/		/E/		/I/		/İ/		/O/		/Ö/		/U/		/Ü/	
	A-		E-		I-		İ-		O-		Ö-		U-		Ü-	
	-A		-E		-I		-İ		-O		-Ö		-U		-Ü	
	-A-		-E-		-I-		-İ-		-O-		-Ö-		-U-		-Ü-	

AÇIKLAMA:

A : "a" kendi başına kelime

/A/ : "a" kendi başına hece

A- : "a" kendisi ile başlayan bir fonem önünde

-A : "a" kendisi ile biten bir fonem arkasında

-A- : "a" kendisini içeren iki fonem arasında

AB or BA : bir fonem

Sözü edilen alt birimler hece, fonem ve ses (phon) olabilmektedir. Sözcükler, her biri kendine özgü niteliği bulunan ve çok sayıda öğelerdir. Ses birim sınırlarının belirlenmesi ve birlikte söylenme etkilerinin göz önüne alınması gibi sorunlar ortaya çıkmaktadır. Fonem tabanlı tanıma işlemlerinde önce alt ses birimleri tanımaya çalışılır daha sonra bu birimler birleştirilerek sözcükler oluşturulur. Fonem: bir dilde bir sözcüğün anlamını diğer bir sözcüğün

anlamından ayırmaya yarayan en küçük ses birimlerine verilen addır. Eğer bir sesi (phon) değiştirmek, ilgili sözcüğün anlamını değiştiriyorsa bu ses fonem olarak anılmaktadır. Ancak bir sesi değiştirmek, ilgili sözcüğün anlamını değiştirmiyorsa fonemden söz edilmemektedir. Fonemlerin belirlenmesi için en küçük çiftten yararlanılmaktadır. Gel ve kel de olduğu gibi, salt tek bir sesi değişik olan sözcükler en küçük çift olarak anılırlar. En küçük çift içinde değişik gösteren ses, anlam değişikliğine neden oluyorsa sözkonusu ses fonemdir. Gel ve kel örneğinde görüldüğü gibi, /g/ ve /k/ Türkçe'deki iki ayrı fonemi oluşturmaktadır. Ses (phon) taban alınarak bir tanım vermek gerektiğinde fonem (anlam ayırıcı özelliği bulunmayan) benzer seslerden oluşan ses kümesi olarak tanımlanır. Hece ise bir nefeste söylenen sesbirimi (2, 7, 15).

Konuşma tanıma uygulamalarında kullanılan sözcük-altı birimler, hece gibi göreceli olarak büyük birimlerden fon ve fonem gibi küçük birimlere, hatta dilbilimsel karşılığı olmayan ve sadece akustik olarak tanımlanmış özel bazı birimlere uzanan geniş bir yelpaze oluşturmaktadır (8). Sözcük-altı birimlerin seçiminde dikkat edilmesi gereken iki kriter tutarlılık (consistency) ve eğitilebilirlik (trainability) olarak sıralanabilir. Tutarlılık sözcük-altı birimin aynı bağlam (context) içinde aynı karakteristiği, yani çok yakın akustik özellikleri göstermesidir (8). Eğitilebilirlik kriteri ise bu birimlerin iyi modellenebilmesi için eğitim süreci içinde yeterli sıklıkta geçmesi ile ilgilidir. Hece, yarım hece gibi büyük sözcük-altı birimler genellikle tutarlıdır ama sayıca çok olduklarından eğitme süreci zordur. Fon ve fonem gibi küçük sözcük-altı birimler ise göreceli olarak daha az sayıda olduklarından eğitme süreci kolay olmakla birlikte tutarlı değildirler (8).

Konuşma tanıma sistemleri istatistiksel modellerle, özellikle SMM ile yüksek başarımlar yakalamışlardır. Bu sistemlerde tanıma işlemini gerçekleştirilmek için kullanılacak olan birim belirlenmelidir (14). Sürekli konuşma ve kelime yakalama sistemleri için en uygun birim üçlüsesler üzerine kurulu olanlarıdır. Böyle sistemlerde amaç sistemi kullananın isteklerine göre değişebilecek yeni kelimelerin sisteme eklenmesi ve başarımın büyük ölçüde değişmemesi

temeline dayanır. Dolayısıyla ayrık kelime tanıma sistemlerinde sistemin başarımını artırmak için kullanılan iki metoddan birisi olan olası her kelimeyi modellemek için ayrı bir SMM kullanma, kullanılacak SMM havuzunu büyüteceğinden, bu da hem sistemin eğitimi için gerekli ses tabanını üstsel bir şekilde genişlemesine neden olup işlem yükünü ciddi manada arttıracağından hem de arama uzayını şişirip sonuç alma süresini uzatacağından uygun değildir. Diğer metod ise kelimeleri modellemek için tek seslileri kullanılır. Bu metot sonuç alma ve eğitim ses tabanının büyüklüğü açısından avantajlı olsa da kelimeleri ayırt etmek için gerekli olan akustik detay seviyesini yeterince sağlamayabilir. Bu da sistemin başarımını olumsuz yönde etkiler. Bu kıstaslar göz önünde bulundurulduğunda anahtar kelime veya geniş dağarcıklı sürekli konuşma sistemleri için üçlüses kullanımının en uygun metod olduğu görülmektedir (14). Türkçe için olası üçlüses sayısı 27000 civarındadır. Bu sayıdaki üçlüses bile arama uzayını ve eğitimi için gerekli olan ses tabanını çok büyüttüğünden üçlüseslerin bir şekilde kümelenmesi veya azaltılması gerekmektedir. İki şekilde kümeleme veya azaltılma yapılabilir: Birinci metodda herhangi bir ses tabanında geçen üçlüsesler bulunur ve akustik açıdan birbirine benzer olanlar kümelenir. İkinci metodda ise dili iyi modellediği varsayılan bir metin grubu üzerinden dilde en sık geçen üçlüsesler bulunur. Birinci metodun getirisi hazır olan – büyük olasılıkla doğal olan- bir ses tabanını kullanmasına dayanmaktadır. Bunun yanında bulunan üçlüseslerin dili iyi modellemesi, bazı üçlüseslerin çok sık bazılarının çok seyrek kullanılması ve üçlüseslerin kümelendirilmesi – öklid uzaklığına dayalı, K-Means kümelendirmesi gibi metodlar kullanılabilir- aşamasındaki değişkenlerin uygun seçilememesi gibi sorunları vardır. İkinci metodun getirileri ise kontrollü bir şekilde dağarcık oluşturulduğundan fonetik açıdan dengeye dikkat edilebilmesi, kullanılan üçlüseslerin dili iyi modellemesi ve kümelemeye ihtiyaç duyulmamasıdır. Başlıca zayıflıkları ise oluşturma evresinin işleminin yoğun emek istemesi ve okunmuş veriye dayanması nedeniyle bazı durumlarda modellemek istenen ortama uyum sağlamamasındandır (14).

Konuşmacıdan bağımsızlık, ses sinyal niteliği, öğrenme yeteneği, sözlük

dağarcığı, dilbilgisi kullanımı gibi özellikler konuşma tanıma sistemine ilişkin önemli özelliklerdir.

Seslerin kayıt edildiği ortam ile kayıt koşullarının niteliği konuşma tanıma sürecini etkileyen önemli bir özelliktir. Ses kayıtlarının yapıldığı ortam, ya özel yalıtılmış, yansımasız bir oda ya da her hangi bir ortam olabilir. Öylesine seçilmiş bir ortamda kaydı yapılan seslerin tanınması, ortamdan kaynaklanan gürültüden dolayı daha zordur. Seslerin kayıt ediliş biçimi ve kayıt kalitesi, en az seslerin kayıt ortamı kadar önemlidir. Kayıt esnasında kullanılan araçlar yüksek kalitede elektronik donanımlar (mikrofon ve yükseltici donanımlar), orta kalitede donanımlar ve telefon gibi özel amaçlı ve koşulların zorlandığı donanımlar olarak sınıflandırılmaktadır. Bu çalışmada mikrofon aracılığıyla program kontrol edilip konuşma tanıma yaptırılacağı için orta kalitede dinamik özellik taşıyan bir mikrofon kullanılmıştır. Kayıt ortamı olarak da çok fazla gürültünün olmadığı ev ve iş yeri seçilmiştir.

Bir konuşma tanıma sistemince tanınabilen sözcüklerin oluşturduğu küme sistem sözlüğü ya da kısaca sözlük olarak anılır. Sözcük altı ses birimleri tabanında tanıma yapan sistemlerde sözlük, ses takımı (phone set) olarak adlandırılır. Sözlük büyüklüğü, genelde tanıma başarısını olumsuz yönde etkilemektedir. Sözlük dağarcığının artırılması işlem gücü gereksinimini ve hata oranını artırmaktadır. Sözlük büyüklüğünü şu şekilde gruplamak mümkündür;

- Küçük dağarcıklı: 100 den daha az sözcük içerenler
- Orta dağarcıklı: 100 -1000 arasında sözcük içerenler
- Büyük dağarcıklı: 1000 -10000 sözcük içerenler ve
- Çok büyük dağarcıklı: 10000 sözcükten fazla sözcük içerenler

Yaklaşık 5000 – 60000 arası sözcükten oluşan dağarcığa geniş dağarcıklı sözlük denilmektedir (20).

Konuşma tanımada üzerinde çalışılacak dilin dilbilgisi yapısından yararlanmak mümkündür. Böylelikle sözcükler dilbilgisi kurallarının özelliklerinden yararlanılarak daha az ögeli kümeler içinde aranmaktadır. Bu ise tanıma başarısını olumlu yönde etkilemektedir. Ancak diğer taraftan da sistemin genel işlem yükünü artırmaktadır. Doğal dil işleme çalışmaları ses tanıma alanında önemli bir yer edinmiştir. Aslında SMM gibi istatistiki konuların temelini oluşturan sonlu durumlu makineler doğal dil işemenin özüdür. O nedenden dolayıdır ki konuşma tanıma çalışmalarında kullanılacak dilin dilbilgisi yapısının iyi anlaşılması gerekmektedir. Bununla ilgili kod çözme çalışmaları yapılmaktadır.

2.2. Konuşma Tanımada Kullanılan Teknikler

Konuşma tanımada kullanılan belli başlı üç teknik vardır. Bunlar;

- Neural Networks (Yapay Sinir Ağları)
- Time warping - dynamic time warping (Dinamik Zaman Eşleştirme)
- Hidden Markov Models (Saklı Markov Modelleri)

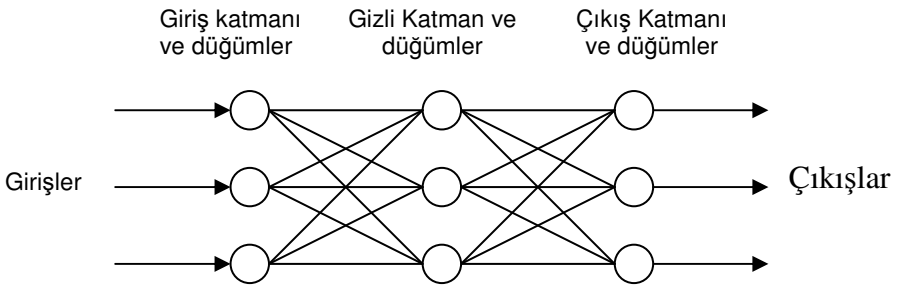
2.2.1.Yapay sinir ağları

Yapay Sinir Ağları; Yapay Zeka' nın konusudur. Yapay Zeka, 1956 yılında John McCarthy tarafından düzenlenen ve Minsky, Newell, Simon, Shannon başta olmak üzere on bilim adamının, iki ay süre ile Dortmouth College' de yaptıkları çalışmaların sonucunda, John McCarthy' in önerisi ile "Artificial Intelligence" ismi ile ilk kez kullanılmış ve yapay zeka bir araştırma disiplini olarak benimsenmiştir (11).

Yapay Sinir Ağı (Artificial Neural Net Model), Connectionist Modeller, Paralel Dağıtılmış İşleme Modelleri veya Neuronorphic System gibi değişik şekillerde isimlendirilmektedir (23). Ancak ismi ne olursa olsun bu modeller, biyolojik sinir sisteminin bilinen yapısını göz önüne alarak, yüksek bir

performansın elde edilmesini sağlayacak şekilde basit hesaplama elemanlarının yoğun bağlantılarından meydana gelmiştir. YSA modelleri, paralel ve yüksek hesaplama hızlarının gerekli olduğu ve mevcut en iyi bilgisayar sistemlerinin dahi gerçekleştirmekten oldukça uzak olduğu, özellikle konuşma ve görüntü algılama başta olmak üzere çeşitli sahalarda büyük bir potansiyele sahiptir (4). Son yıllarda yapay sinir ağları kullanılarak yapılan çalışmalar oldukça başarılı sonuçlar vermiştir (24). İnsan beyninin çalışma prensibi üzerine oturtulmuş olan YSA' lar giriş ve çıkış veri kümelerini kullanarak sistem davranışını öğrenebilen yapay sistemlerdir (31).

Şekil 2.4 de genel bir YSA modeli görülmektedir. YSA kullanarak sinyal tanıma 1950' li ve 1960'lı yıllarda oldukça pahalı bir araştırmayken, 1969' dan 1982' ye kadarki dönemde bu alandaki pahalılık giderek azalmıştır. Üstelik YSA ile ilgilenen bir çok yeni alan ortaya çıkmıştır ve konuşma tanıma da bu alanlardan birisidir (12).



Şekil 2.4. Genel bir yapay sinir ağı modeli (3)

Konuşma tanıma sistemlerinde YSA yaklaşımına dayalı çeşitli yöntemler geliştirilmiştir (32). Bunlar;

- Perceptron
- Multilayer Networks
- Backward Error Propagation
- Kohonen Self-Organizing Maps
- Hopfield Nets ve Associative Memory

YSA' ları konuşma tanıma alanına uygun bir tekniktir. Çünkü konuşma tanımada kesin sınırlar belli değildir. YSA' larının dezavantajları da bulunmaktadır. Bu teknikteki en büyük problem karmaşık sorunları çözmek için (konuşma tanıma gibi) ya çok büyük ya da çok katmanlı ve çok nöron içeren sinir ağlarına ihtiyaç duyulmaktadır. YSA' lar büyüdükçe çalışmaları üstel bir şekilde yavaşlamaktadır. Bu problemi de paralel işleme teknolojisiyle çözmek mümkündür. YSA' ların eğitme ve kullanma olmak üzere iki türlü çalışma şekli vardır. Eğitme aşamasında YSA' ndaki düğümlerin birbirine bağlantı yüzdelerini gösteren ağırlık değerleri hesaplanır. Eğitme aşamasında elde edilen bu ağırlık değerleri daha sonra sadece girişlerin verilip çıkışların hesaplanmasının istenildiği kullanma aşamasında işe yararlar. Kullanma aşamasındaki algoritma eğitme aşamasına göre daha kolaydır. Bundan dolayı da YSA' lar bu aşamada daha hızlı çalışırlar. Eğitme aşamasında karşılaşılan sorunlar bu aşamada oluşmaz.

YSA' ları konuşma tanıma sistemlerinde şu şekilde kullanılabilir. Örneğin; belirli bir kelimeden hızlı fourier dönüşüm, doğrusal önkestirim kodu veya kod etkileşimi doğrusal önkestirim ile ya da başka bir yöntemle elde edlien katsayılar, YSA' nın giriş katmanına yüklenir. Kullanılan teknik ve elde edilen katsayıların miktarı, YSA' nın başarısını ve çalışma hızını etkilemesi açısından önem taşımaktadır. Sonra çıkış katmanına bu kelimeyi temsil edecek bir kod yüklenir. Bu kod bu çalışmayı yapacak olan kişinin belirlediği bir teknik olabilir. Ancak genel olarak 0..1 veya -1..1 arasındaki değerler bu tekniğe uygundur. Örneğin 0,5 ten büyük olan çıkışlar 1 kabul edilip diğerleri 0 kabul edilerek ikili kodlama yapmak mümkündür. Ya da 1' e en yakın çıkışın numarasını kullanmak tercih edilebilir. Sonra YSA' nı eğitmek için hangi algoritma kullanılıyorsa çalıştırılarak işleme devam edilir. Başka bir kelime için yine bu çalışmalar tekrar edilir. Eğitme aşaması yeterli görülene kadar sürmelidir. Yeterli olduğunun anlaşılması için oluşan toplam hatanın belli bir yüzdenin altına inmesi gerekmektedir. Eğitme aşamasında kullanılan verilen sırası rastgele olmalıdır. Eğer benzer karakterdeki veriler öbek halinde eğitme işlemine uygulanırsa, öğrenme en son öbek için daha iyi olabilir. Bu durumda doğru bir çalışma değildir. Çünkü eğitme aşamasındaki

döngünün içinde sıra rastgele seçilmemiştir. Ayrıca bazı durumlarda öğrenmenin gerçekleşmeyeceği de düşünülerek döngünün sonlandırma şartına belirli bir tekrar sayısının aşılması koşulu da eklenmelidir. Bu aşamadan sonra kullanma aşamasına geçilir. Bu aşamada konuşulan bir kelimenin hesaplanan katsayıları verilerek çıkışların hesaplanması için YSA' ları çalıştırılır. Programdan elde edilen çıkış kodlarına göre de hangi kelimenin konuşulduğu anlaşılmaya çalışılır.

2.2.2. Zaman eşleştirme

Zaman eşleştirme yöntemi konuşma tanıma yöntemlerinde sıklıkla kullanılan bir diğer yöntemdir. Bu yöntem daha çok diğer yöntemlerle birlikte kullanılan ve daha çok tanıma işlemlerinin verimliliğini artırmak amacıyla kullanılan bir yöntemdir. Bu yöntemde, konuşma ifadelerini seslendirme süreleri sıkıştırılarak ya da genişletilerek referanslarla karşılaştırılmaları ilkesi kullanılmaktadır (7). Aynı sözcüğü aynı kullanıcı tekrar seslendirdiğinde bile bir seslendiriliş daha önceki seslendirilişlere benzemeyebilir. Sözcüğün uzunluğu doğrusal olmayan bir biçimde genişleme ve daralma gösterir (26).

Zaman eşleştirme yöntemi sözcüğün ya da fonemin sinyalinin, referans şablonu ile aynı zaman aralığında olabilmesi için zaman ekseninde daralma yada genişleme yapmayı amaçlar. Sözcük tanıma ya da fonem tanıma için genel olarak dinamik zaman eşleştirme yöntemi kullanılmaktadır. Dinamik zaman eşleştirme yönteminde zaman ekseni doğrusal olmayan bir biçimde genişletilip daraltılarak referans şablonu ile tanınacak olan sesli ifade kesiminin başlangıç ve bitiş zamanları çakıştırılmaya çalışılır. Amaç karşılaştırmanın aynı zaman aralıkları için yapılmasını sağlamaktır. Dinamik zaman eşleştirme işlemi, devingen programlama tekniği kullanılarak gerçekleştirilir (26).

Zaman eşleştirme işleminde sorun A ve B örneklerinin çakıştırılmasıdır. Devingen programlamanın uygulanışı için iki örnek zaman dizisi düşünüldüğünde;

karşılaştırılacak iki örneğin zaman eksenindeki değerleri,

$A = a_1, a_2, \ldots, a_i, \ldots, a_m$
$B = b_1, b_2, \ldots, b_j, \ldots, b_n$

olarak tanımlanmışsa zaman eşleştirme fonksiyonu:

$C = c(1), c(2), \ldots, c(k), \ldots, c(K)$ olarak yazılabilir.

Burada c, örneklerin kesişen nokta çiftlerini vermektedir. Bunun için devingen programlama yöntemi kullanılarak bir fonksiyon tanımlanır. Bu fonksiyon iteratif bir yaklaşımla sesli ifadeyi daraltarak ya da genişleterek referans şablonu ile aynı zaman aralığına getirir.

2.2.3. Saklı markov modelleri

Markov zincirleri olarak da bilinir ve olasılık kuramının çok önemli ve iyi çalışılmış bir kavramıdır (19). Saklı Markov Modelleri stokastik prosesleri modelleyebilen sonlu durum ağlarıdır. Her durum, gözlem vektörü uzayına farklı bir bölge ve karakteristiği tanımlar. Durumlar arası geçişler de modellenen öznitelik değişimlerini ele alır. Durumlar içerisinde verilen herhangi bir öznitelik gözlem vektörünün duruma uygunluğunu veren çıktı olasılık dağılım fonksiyonları yer alır.

Bu yaklaşım ilk olarak 1965 - 70 yıllarında kullanılmaya başlanmış ve 1985 - 90 yıllarında sesli ifade tanımada çok kullanılan bir yöntem olmuştur. SMM yönteminin popülaritesinin artmasının iki temel nedeni vardır (30). Bu sebeplerden birincisi sahip olduğu zengin matematiksel yapıdır. Bu yüzden çok çeşitli uygulamalarda kullanılabilmesi için bir teorik temeli vardır. Diğer neden ise, uygun bir biçimde uygulandığında, SMM yöntemi, başarılı sonuçlar elde edilmesini sağlamaktadır. Bir araştırmaya göre SMM ile yapılan konuşma tanıma sistemleri diğer tekniklerle de başarı sağlamasına rağmen en başarılı sonuçlar veren teknik olmuştur (28). SMM bir sinyalin stokastik

olarak modellenmesidir. Sesli ifade tanımada kullanılması kısaca, ardışık kısa süreli sesli ifade kesimlerinin birlikte ele alınması ile ardı ardına gelebilecek bu kesimler için bir model oluşturmak ve bu modelden yararlanarak uzun süreli sesli ifade kesimlerinin tanımasını sağlamak şeklinde özetlenebilir (30). Markov işlemi yada zinciri kesikli bir *t* zamanındaki N durumdan biridir. Durum değişkeni *qt* ile belirtilir. Genelde sistemin tam bir tanımı önceki durumların sırasının bilinmesine bağlıdır. Birinci dereceden bir Markov işleminde sistemin şu andaki durumu sadece bir önceki duruma bağlıdır.

Durum geçiş olasılıkları zamandan bağımsız olduğundan dolayı, bir Markov işlemi durum geçişleri ile tanımlanabilir (26). Durum geçis matrisi Eş. 2.1' deki gibidir.

$$A = \lfloor a_{ij} \rfloor,$$
$$a_{ij} = P(q_t = j / q_{t-1} = i), i, j = 1,..., N \qquad [2.1]$$

Buradaki olasılıklar Eş. 2.2' deki sınırlamayı sağlamak zorundadır.

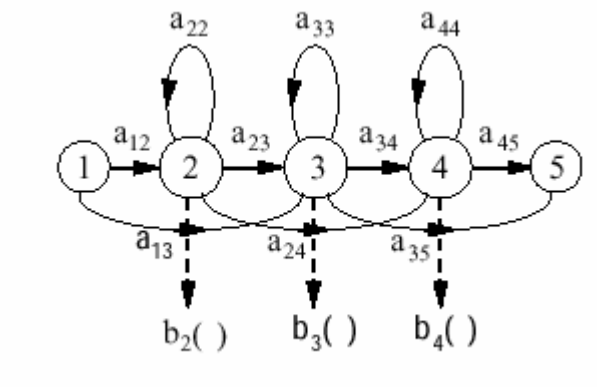
$$\sum_{j=1}^{N} a_{ij} = 1,$$
$$a_{ij} \geq 0. \qquad [2.2]$$

Sistemin ilk durumu q_0 olarak tanımlanır ve bu durum ilk değer olasılıklar vektörü $\pi = [\pi_1, \pi_2, ..., \pi_N]^T$ ile belirlenir (26). Bundan sonra gelecek olan herhangi bir durum dizisi $q = (q_0, q_1, ..., q_r)$, için bir Markov işlemi tarafından üretilebilme olasılığı Eş. 2.3' deki gibi tanımlanır.

$$P(q \mid A) = \pi_{q_0} a_{q_a q_1}, a_{q_1 q_2}, ..., a_{q_{T-1} q_T} \qquad [2.3]$$

Bir Markov işlemi için eğer durum dizisi q doğrudan gözlenemiyorsa “gizli” olarak adlandırılır. Bu durumda sadece durumların dolaylı olarak gözlemlenmesi söz konusudur.

Bir gözlem ve bir durum arasında her ne kadar birebir bir gereklilik yoksa da her bir durumun belirli bir olasılıkla gözlenmesi gerekmektedir (Şekil 2.5).



Şekil 2.5. Basit bir soldan sağa SMM (34)

Sesli ifade tanıma bağlamında gözlemler özellik vektörleridir. SMM durumları ise temel alınan sesli ifade birimlerine denk gelir. Bu durumda amaç saklı olan durum dizisini gözlemlerden yararlanarak bulmaktır (26). Fonem tabanlı bir sesli ifade tanıma sisteminde bulunan durum dizisi fonem dizisine denk gelir. Her fonem için ayrı bir model tanımı yapılır.

Her bir fonem için böyle bir model oluşturulduğu düşünülürse ardarda gelen fonem zincirleri bu modellerin ardarda sıralanması ile modellenebilir. Bu durumda her bir fonemin son durumundan bir sonraki fonemin ilk durumuna bir geçiş söz konusudur (26).

Böylece tüm sesli ifade tanıma sistemi bir Markov işlemi olarak tanımlanabilir. Böyle bir model farklı uzunluktaki fonem dizilerini tanımak için kullanılabilir. Saklı Markov Model yaklaşımı daha biçimsel tanımlanacak olursa; herhangi bir *t* anında yapılan bir gözlem O_t ile belirtilsin (26). Herhangi bir *i* durumu tarafından O_t' nin üretilmesi olasılığı Eş. 2.4' de belirtilmiştir.

$$b_i(O_t) = P(O_t \mid q_t = i) \quad [2.4]$$

$b_i(O_t)$ şeklinde tanımlanmış olasılıkların kümesi $B = \{b_i(O_t)\}_{i=1}^{N}$ olarak tanımlanır. Durumlar tarafından üretilen gözlemlerin birbirinden bağımsız olduğu varsayılır ve bir durum dizgisinin ürettiği gözlemler $O = (O_1, O_2, ..., O_T)$ şeklinde ifade edilecek olursa bu gözlemlerin olasılığı Eş. 2.5 deki gibidir.

$$P(O \mid \mathrm{q}, \mathrm{B}) = b_{q_1}(O_1) b_{q_2}(O_2), .., b_{q_T}(O_T) \quad [2.5]$$

son olarak gözlem dizisi O ve durum dizisi q dizilerinin birleşik olasılıkları da bu ikisinin olasılıklarının çarpımıyla bulunur.

$$P(O, q \mid \pi, A, B) = \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1}} b_{q_t}(O_t) \quad [2.6]$$

Bu durumda bir HMM, $\lambda = (\pi, A, B)$ şeklinde ifade edilebilir.

SMM geliştirme aşaması üç alt probleme ayrılabilir (15, 26). Bunlar;

- Hesaplama problemi; verilen bir model için belirli bir gözlem dizisinin olasılığının hesaplanması
- Çözümleme problemi; verilen bir model için belirli bir gözlem dizisini oluşturan durum dizisinin bulunması.
- Öğrenme problemi; verilen bir model için bir dizi gözlemi oluşturan bir modelin olasılığının yüksek olması için model parametrelerinin kararlaştırılması.

(15) SMM' ye basitce bir örneği şu şekilde vermiş; "örneğin, bir öğrencinin genellikle ya kafede, ya okulda olabildiği durumunu göz önüne alalım. Eğer

öğrencinin birkaç hafta aktivitesini gözlemlersek şöyle bir modelle karşılaşırız.

Kafe → okul → kafe → uyku → kafe → uyku → okul → kafe → uyku → kafe

Böylece belli bir süre boyunca öğrenciyi on kez gözlemlemiş oluruz. Öğrencinin okuldan kafeye gitme olasılığını bulabiliriz (bu yüksek bir olasılık çünkü her seferinde öyle yapmış). Aynı zamanında öğrencinin okuldan uykuya gitme olasılığını hesaplayabiliriz (pek olası değil çünkü hiç yapmamış).

Bunu temsil etmenin diğer yolu da Çizelge 2.3 deki gibi 0 (imkansız) dan 1 (mutlaka) kadar bir ölçek ile temsil edilmiştir.

Çizelge 2.3. Saklı Markov Model örneği

	Uyku	okul	Kafe
Uyku	0	0,3	0,2
Okul	0,1	0	0,8
Kafe	0,9	0,7	0

Öğrencinin davranışını tahmin etmek için bu diyagramı kullanabiliriz. Eğer öğrenci kafedeyse gitmesi en muhtemel yer okuldur. Çünkü kafede en yüksek sayı 0,8 dir. Eğer öğrenci okuldaysa (yine üst sırada) o zaman gitmesi en muhtemel yer kafedir. Çünkü en yüksek değere sahip (0,7). Bu yöntem Saklı Markov Modeli olarak bilinir.

Bir diğer tarafından takip edilen sesin frekansına bakarak sesler için bir SMM model oluşturabiliriz. Ve bunu bir sonraki en muhtemel sesi tahmin etmekte kullanabiliriz. Bu bizim olası olmayan adayları elememize ve konuşulan cümle için olası ses kümesini ortaya çıkarmamıza yardımcı olur.“

Şekil 2.6 da basit bir şekilde oluşturulmuş bir SMM tanımı görülmektedir.

```
~h "hmm1"
<BeginHMM>
    <VecSize> 4 <MFCC>
    <NumStates> 5
    <State> 2
       <Mean> 4
          0.2 0.1 0.1 0.9
       <Variance> 4
          1.0 1.0 1.0 1.0
    <State> 3
       <Mean> 4
          0.4 0.9 0.2 0.1
       <Variance> 4
          1.0 2.0 2.0 0.5
    <State> 4
       <Mean> 4
          1.2 3.1 0.5 0.9
       <Variance> 4
          5.0 5.0 5.0 5.0
    <TransP> 5
       0.0 0.5 0.5 0.0 0.0
       0.0 0.4 0.4 0.2 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>
```

Şekil 2.6. Basit bir soldan sağa SMM tanımı (34)

Viterbi algoritması

Viterbi algoritması genel anlamıyla bir dinamik eşleştirme (dynamic warping) yöntemidir. SMM için kullanımı, tanınacak sese ait öznitelik gözlem vektör sekansının modele ait durumlara dağıtılarak en uygun durum-gözlem vektörü eşleşmesini bulmak ve buna ait olurluk değerini hesaplamak olarak gerçekleşir (9).

Sürekli yoğunluklu Gauss karışımları

İstatistikte en yoğun kullanılan olasılık dağılım fonksiyonu Gauss dağılımıdır.

İstatistiksel konuşma tanımada SMM durumları, Gauss dağılımları ile ifade edilebilir. Eğitim verisinin durumuna göre birden çok sayıda Gauss dağılımının ağırlıklı toplamı ile sürekli yoğunluklu Gauss karışımları (Continuous Density Gaussian Mixtures) elde edilir (34).

Model parametre tahmini (parameter estimation)

Parametre tahmini işlemi SMM' lerin eğitilmesidir. Bir SMM' nin eğitilecek parametreleri şunlardır:

- Her durum için olasılık dağılım fonksiyonunun parametreleri. Gauss karışımları için bu parametreler: karışım sayısı, karışım ağırlıkları, karışım ortalamaları ve kovaryansları.
- Durumlar arası geçiş tablosu olasılık değerleri.

Baum-Welch algoritması SMM parametre tahmini işlemini maksimum olurluk yaklaşımı (maximum likelihood sense) ile çözen bir en iyileştirme tekniğidir. Ayrıca HTK tarafından kullanılan gömülü yeniden tahmin yöntemi (embedded re-estimation) geniş veritabanları üzerinden Baum-Welch uygulanmasına olanak tanır (34).

Sınırlı dağarcıklı ayrık sözcük tanıma

SMM uygulamalarının ilk örneğidir. Sistem dağarcığı sınırlı sayıda sözcükten oluşur. Tanınacak konuşma da bu sözcüklerden herhangi birinin bir kez söylenmesinden ibarettir. Bu tür uygulamalarda her sözcük için bir SMM modellenir. Tanıma işlemi ses sinyalinin bu modellerden geçirilerek olurluk değerlerinin hesaplanması ve en yüksek olurluk değerini veren modelin tanınan sözcük olarak seçilmesi şeklinde gerçekleştirilir.

Geniş dağarcıklı tanıma

Geniş dağarcıklar söz konusu olduğunda her sözcük için bir SMM üretilmesi pratik olmaktan çıkmaktadır, çünkü:

- Her sözcük için ayrı ayrı eğitim verisi gerekmektedir.
- Sistem dağarcığına yeni bir sözcük eklenmesi için o sözcüğe ait eğitim verisi bulunması ve yeni bir model üretilmesi gerekecektir.
- Sürekli konuşma tanıma için sözcük modelleri uygun değildir çünkü bir sözcüğün ayrık ve sürekli konuşma içindeki söylenişleri farklıdır.

Bu nedenle SMM' ler sözcükler yerine sözcük-altı birimler, örneğin fonem, için oluşturulur. En yaygın kullanılan seçim bağlam bağımlı modeller kullanmaktır. Burada da üçlü-ses modelleri (tri-phone) karşımıza çıkmaktadır.

2.3. HTK Araç Kitinin Özeti

Konuşma tanıma sistemimizde kullanılan akustik modellerin üretimi Cambridge Üniversitesinde geliştirilen "Hidden Markov Toolkit" (HTK) adlı bir araç kiti sayesinde gerçekleştirilmiştir. Bu sistem konuşma tanıma işlemleri için gerekli olabilecek çeşitli işlevleri (özellik çıkarımı, eğitim, tanıma, değerlendirme) yerine getirebilecek birimlerden oluşmuştur. Bu sistemin kullanılmasında çeşitli nedenler vardır: Sistem eğitim amaçlı kullanımlarda limitsizdir ve açık kaynak koduna sahiptir. Ayrıca ciddi bir kullanıcı kitlesine sahip olup çeşitli ortamlar için başarımı irdelenmiştir. Sistem optimal olmayabilir ve kendisinden çeşitli açılardan daha iyi tanıma sistemleri var olabilir ancak yukarıda verilen sebeplerden ötürü kullanımı tercih edilmiştir.

HTK, ağırlıklı olarak konuşma tanıma sistemleri için gerekli olan SMM' lerin oluşturulması ve daha sonra oluşturulan SMM' ler ile tanıma ve değerlendirme için kullanılabilecek çeşitli araçların bir araya geldiği bir yazılım paketidir. Sistem metin bazlı komutlarla yürütülür. Bundaki

amaçlardan biri çok çeşitli seçenekler denenirken kullanılmış olunan komut dizisinin saklanmasının kolaylığıdır.

Sistem öncelikle teklisesler için elle zaman-etiketlenmiş az miktarda ilk koşul bilgisi ister. Bunlar ve viterbi algoritması kullanılarak kaba bir model elde edilir. Daha sonra sistem tabanında geri kalan sadece etiketlenmiş (zaman-etiketine ihtiyaç yok) veriler ve Baum-Welch kullanılarak daha detaylı bir şekilde tekseslilер için SMM' leri eğitilir (14).

Eğitim ve tanıma işlemleri dışında konuşma tanıma sistemi için gerekli olan değerlendirme ve sinyal işleme ön uç birimi işlemleri sunan modülleri de vardır. Değerlendirme modeli için dinamik programlama metodu kullanılır. Sinyal işleme ön uç birimi ise ham ses verisinin çeşitli biçimlerde ifade edilmesine olanak tanır, bunlardan bazıları doğrusal öngörü kodlaması (DÖK) ve MFCC' dir (14).

2.3.1. HTK yazılım mimarisi

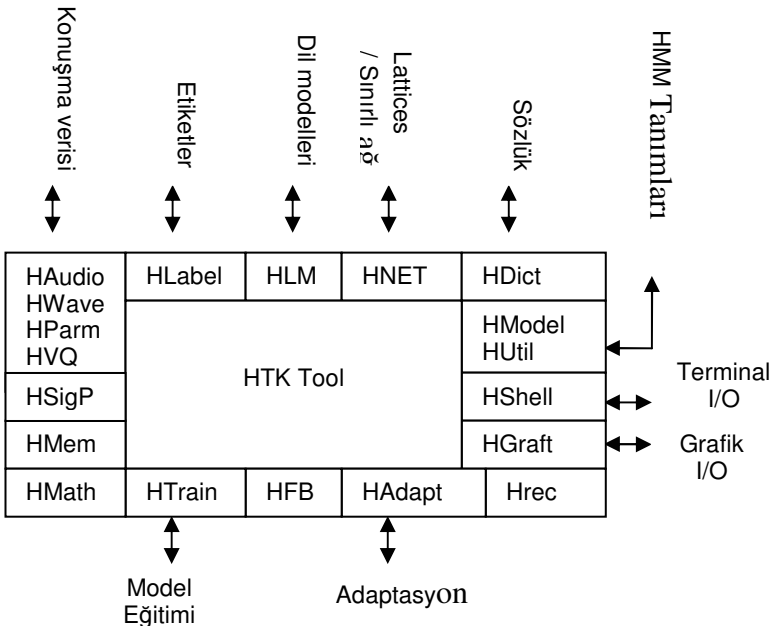
HTK' nın işlevselliğinin büyük kısmı kütüphane modüllerinden kaynaklanır. Bu modüller, tüm araçların aynı şekilde dış dünyayla kesişmesini sağlar. Bunlar ayrıca ortaklaşa kullanılan işlevlerin merkez kaynağını oluşturur.

Kullanıcı, girdi/çıktı ve işletim sistemiyle etkileşim, kütüphane modülü HSHELL tarafından kontrol edilir ve hafıza yönetiminin tümü HMEM tarafından kontrol edilir.

Matematik desteğini HMATH sağlar ve konuşma analizi için gerekli sinyal işleme işlemleri HSIGP' dedir. HTK' nın gerektirdiği dosya türlerinin tümünün kesişme modülü vardır.

HLABEL, etiketlendirme dosyaları, HLM dil modeli dosyaları, HNET ağlar, HDICT sözlükler, HVQ, VQ kod kitapları ve HMODEL HMM tanımları için kesişme sağlar (34).

Bu dosyalara yönelik olarak Şekil 2.7 de tipik bir HTK aracının yazılım yapısı ve bunun girdi / çıktı kesişmesi gösterilmektedir.



Şekil 2.7. HTK yazılım mimarisi

Dalga formu düzeyinde konuşma girdi ve çıktılarının tümü HWAVE yoluyla olur ve parametrik düzeyde bu HPARM yoluyla gerçekleşir. Uyumlu bir kesişme sunan HWAVE ve HLABEL verilerin diğer sistemlerden alınmasını sağlayarak çoklu dosya formatını destekler. Doğrudan işitsel girdi HAUDIO tarafından desteklenir ve basit etkileşimsel grafikler HGRAF tarafından sağlanır. HUTIL, HMM' lerin elde edilmesi için kullanım yönleri sunar. Öte yandan HTRAIN ve HFB HTK eğitim araçlarına destek sağlar. HADAPT, çeşitli HTK adaptasyon araçları için destek olur. Son olarak HREC temel tanıma işleme işlevlerini kapsar (34).

Bu kütüphane modüllerinin davranışlarının kontrolü, konfigürasyon değişkenlerinin oluşturulmasıyla sağlanır.

2.3.2. HTK aracının genel özellikleri

HTK araçları geleneksel emir-hat tarzı kesişme yoluyla çalışacak şekilde tasarlanmıştır. Her bir aracın bir dizi gerekli argümanı ve seçimlik argümanı bulunur. Seçimlik argümanlar her zaman eksi işaretiyle belirtilir. Örneğin aşağıdaki komut HFOO adı verilen HTK aracını işletir.

HFOO –T 1 –f 34.3 -a -s myfile file1 file2

Bu aracın file1 ve file2 adı verilen iki temel argümanı vardır. Bunlara ek olarak seçimlik 4 argümanı bulunur. Seçenekler her zaman uygun olduğunda seçenek değerini izleyen tek bir harflik seçenek adıyla sunulur. Seçenek değeri her zaman boşluk bırakılarak seçenek adından ayrılır. Dolayısıyla –f seçeneğinin değeri gerçek sayıdır. –T seçeneğinin değeri integer sayıdır ve –s seçeneğinin değeri bir dizidir. –a seçeneğinin değeri yoktur ve bu sadece aracın bazı özelliklerini etkin kılmak ya da etkinliğini ortadan kaldırmak için basit bir bayrak olarak kullanılır. Adları büyük harf olan seçeneklerin tüm araçlarda anlamı aynıdır. Örneğin –T seçeneği, bir HTK aracının iz çıktısını kontrol etmek üzere kullanılır (34).

Komut satır argümanlarına ek olarak bir aracın işlemesi konfigürasyon dosyasındaki parametrelerle kontrol edilebilir. Örneğin if komutu.

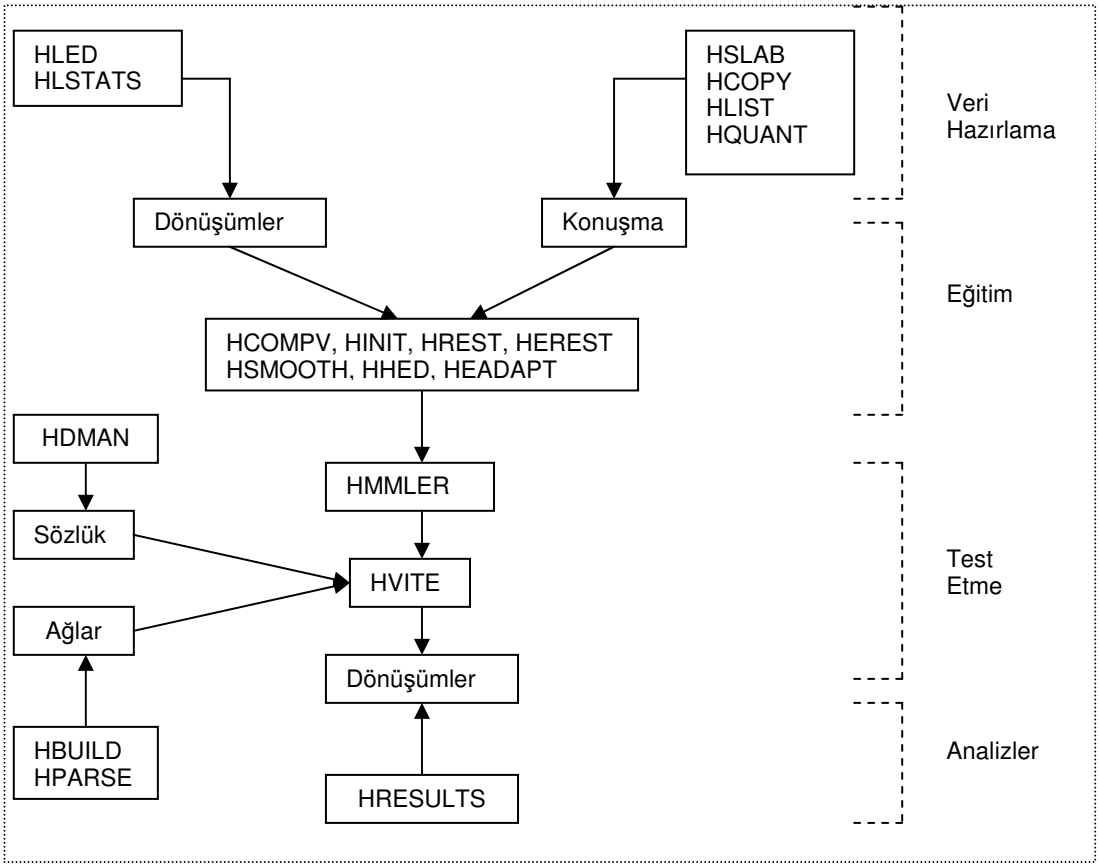
HFOO –C config1 –C config2 –f 34.3 -a -s myfile file1 file2

İse HFOO aracı başlangıç işlemleri sırasında konfigürasyon dosyası config' de saklanan parametreleri yükler. Çoklu konfigürasyon dosyaları –C seçeneği tekrar edilerek nitelendirilebilir. Konfigürasyon parametreleri komut satırı argümanlarına alternatif olarak kullanılabilir. Örneğin iz seçenekleri her zaman konfigürasyon dosyasında oluşturulabilir. Ancak konfigürasyon dosyalarının temel kullanımı tüm HTK araçlarının dayandığı kütüphane modüllerinin ayrıntılı davranışlarını kontrol etmektir.

Bu tür komut satır işlemleri modern kesişmelerle karşılaştırıldığında garip görülebilir ancak bunların bazı avantajları vardır. HTK araç uygulamasını kontrol etmek üzere kabuk (shell) senaryolarının yazımını kolaylaştırır. Bu büyük ölçekli sistem oluşturma ve denemede önemlidir. Daha da ötesi metin temelli komutları kullanarak tüm işlemlerin tanımlanması sistem yapısının veya deneysel işlemin ayrıntılarının kaydedilip belgelenmesini olanaklı hale getirir (34). Son olarak, komut satır özeti ve HTK aracının seçenekleri argümanı olmayan aracın uygulanmasıyla da elde edilebilir.

2.3.3. Araç kiti

Sözcük-altı temelli sürekli konuşma tanıyıcısı oluşturma dört temel evrede gerçekleşir: Veri hazırlama, eğitim, ölçme ve analiz (Şekil 2.8).



Şekil 2.8. HTK işleme durumları

Veri hazırlama araçları

SMM' leri oluşturmak için bir dizi konuşma verisi dosyası ve bunlarla ilgili dönüşümler gereklidir. Çoğunlukla konuşma verileri veri tabanı arşivlerinden elde edilir. Eğitimde kullanılmadan önce bunlar uygun parametrik forma dönüştürülür ve ilgili dönüşümler da bu formatta hazırlanır ve gerekli sözcük adları kullanılır. Konuşmanın kaydedilmesi gerekirse HSLAB aracı hem konuşmanın kaydedilmesi hem gerekli dönüşümlerle işaretlenmesi için kullanılabilir.

Pratikte verilerin parametrize edilmesi faydalıdır. Bunun için HCOPY aracı kullanılır. Bu araç bir veya daha çok kaynak dosyasının çıktı dosyasına kopyalanması için kullanılır. HCOPY tüm dosyayı kopyalar ancak bir dosyadan belli bölümün alınması için çeşitli mekanizmalar vardır. Uygun konfigürasyon değişkenlerinin hazırlanmasıyla tüm girdi dosyaları parametrik forma dönüştürülebilir. Dolayısıyla bu şekilde bir dosyanın kopyalanması istenen kodlamayı gerçekleştirir. HLIST aracı, çok miktarda verinin işlenmesinden önce dönüştürme sonuçlarını kontrol etmek için kullanılabilir. Dönüşümlerinde hazırlanması gerekir. Tipik olarak orijinal kod dönüşümlerde kullanılan adlandırmalar istenildiği gibi olmayacaktır. Ayrıca HMM eğitimi adlandırmaların bağlama dayalı olmasını gerektirir (34).

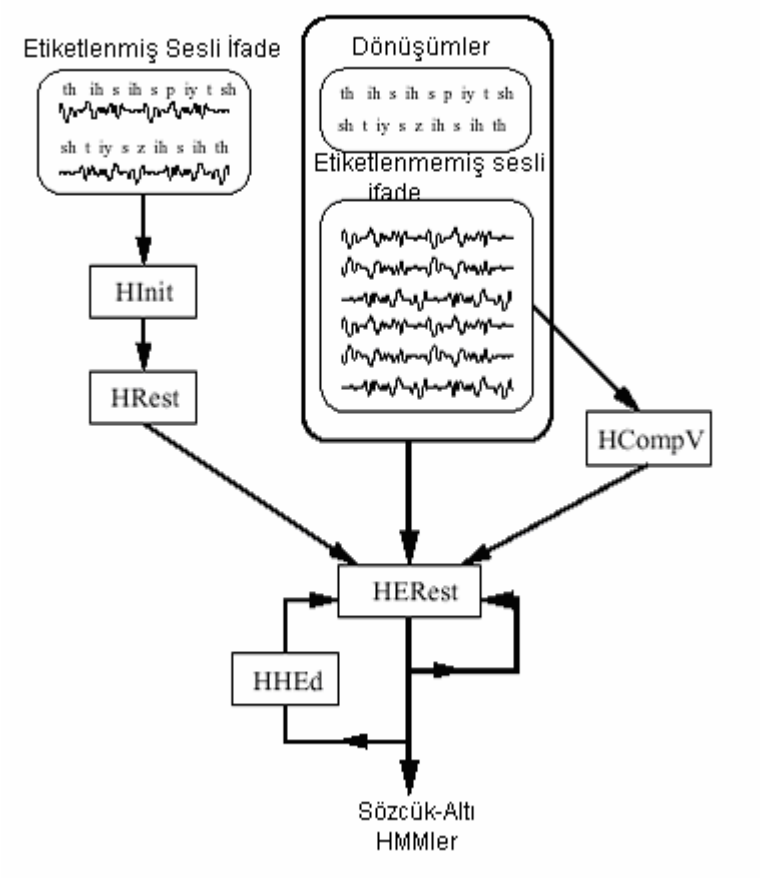
HLED aracı senaryo temelli bir adlandırma editörüdür ve dosyaların adlandırılması için istendik dönüşümlerin yapılması için tasarlanmıştır. HLED ayrıca işleme için daha yerleşik olan Master Label Dosyasına (MLF) dosyaları aktarır. Veri hazırlamada son olarak HLSATS ad dosyalarındaki istatistiği toplar ve gösterir. Gerektiğinde HQUANT farklı olasılık SMM sistemi oluşturmak için hazırlamada VQ kod kitabı oluşturmak amacıyla kullanılabilir.

Eğitim araçları

Sistem oluşturmanın ikinci adımı, prototip tanım yazarak her bir SMM için gerekli topolojinin tanımlanmasıdır. HTK, SMM' lerin istenen topolojiye sahip

olarak oluşturulmasını sağlar. SMM tanımları basit metin dosyaları gibi dışsal olarak saklanabilir ve dolayısıyla bunların herhangi bir metin editörüyle düzenlenmesi mümkündür. Alternatif olarak standart HTK dağılımında, en yaygın topolojileri otomatik olarak üretmek için örnek SMM prototipleri ve senaryo bulunur. Geçiş olasılıkları dışında prototip tanımında verilen tüm SMM parametreleri göz ardı edilir. Prototip tanımın amacı sadece SMM özelliklerini ve topolojisini belirtmektir. Aktüel parametreler daha sonra eğitim araçları tarafından hesaplanır. Geçiş olasılıkları için duyarlı değerler verilmelidir. Ancak eğitim süreci bunlara son derece duyarsızdır. Bu olasılıkların seçilmesine yönelik kabul edilebilir ve basit bir stratejisi olası durumun geçişlerinin tümünü yapmaktadır.

Aktüel eğitim sürecinin evreler halinde meydana geldiği durum Şekil 2.9 da verilmektedir. İlk olarak modellere ilişkin başlangıç dizisi yaratılmaktadır. Sözcük-altı sınırının yeri işaretlenen konuşma verileri varsa bu, bootstrap veri olarak kullanılabilir. Bu durumda HINIT ve HREST araçları tamamen adlandırılan bootstrap verileri kullanılarak ayrık sözcük tarzı eğitim sunar (34).



Şekil 2.9. Sözcük-Altı SMM' lerinin eğitimi

Gerekli SMM' lerin her biri tek tek üretilir. HINT tüm bootstrap eğitim verilerini okur ve istenen ses birimin örneklerinin tümünü çıkarır. Daha sonra bölümsel k-ortalama işlemini kullanarak parametre değerlerinin başlangıç dizisini sürekli olarak hesaplar. İlk döngüde eğitim verileri tek bir şekilde bölümlenir, her bir model durumu eş veri bölümleriyle eşleştirilir ve daha sonra ortalama ve varyans tahmin edilir. Karışım Gauss modelleri eğitiliyorsa k-ortalama ve varyans tahmin edilir. Karışım Gauss modelleri eğitiliyorsa k-ortalama öbeklemenin değişik bir formu kullanılır. İkinci ve ardışıl döngülerde tek tip bölümlemenin yerini Viterbi düzenleme alır. HINIT tarafından hesaplanan başlangıç parametre değerleri HREST tarafından yeniden tahmin edilir. Yine tamamen adlandırılan bootstrap veriler kullanılır. Ancak bu kez bölümsel k-ortalama işleminin yerini Baum-Welch yeniden tahmin işlemi alır. Bootstrap veriler yoksa düz başlangıç kullanılabilir. Bu durumda ses birimi (phone,fon) modellerinin tümü eş olacak şekilde başlatılır ve global konuşma ortalaması

ve varyansına eş durum ortalamaları ve varyanslarına sahiptir. Bunun için HCOMPV aracı kullanılır (34).

Modellerin başlangıç dizisi yaratıldıktan sonra HEREST aracı tam eğitim dizisi kullanılarak yerleştirilmiş eğitimin uygulanması için kullanılır. HEREST eş zamanlı olarak SMM ses birimi modellerinin tüm dizisinin tek bir Baum-Welch yeniden tahminini yapar. Her bir eğitim sözcesi için eş ses modelleri oluşturulur ve daha sonra ileri-geri algoritma, bölümdeki her bir SMM için alan işgali ortalamalar, varyanslar hakkındaki istatistiği artırmada kullanılır. HEREST temel HTK eğitim aracıdır. Bu, büyük veri tabanlarını işlemek üzere tasarlanmıştır. Bu hesaplamanın azaltılmasında kolaylık sağlar ve makine ağlarında paralel çalıştırılabilir.

HTK da sistem yapılandırmanın felsefesi şudur: SMM' ler tekrar ayrıştırılmalıdır. Dolayısıyla tipik ilerleme, tekil Gauss bağlama dayalı olmayan ses birimi modellerinin basit dizisiyle başlamaktır. Daha sonra bağımlılığını içermek üzere bunlar genişletilerek ayrıştırılabilirler. HHED aracı HMM tanım editörüdür. Genel süreç HHED kullanılarak evreler halinde SMM' ler dizisinin değiştirilmesi ve her evreden sonra HREST kullanılarak değiştirilen dizinin parametrelerini yeniden tahmin etmektir. Belli konuşmacıların performansını iyileştirmek üzere HEADAPT VE HVITE araçları kullanılabilir. Son ürün konuşmacıya uyarlanan sistemdir (34).

Bağlama bağımlı SMM sistemleri oluşturmanın sorunlarından biri, verilerin yetersiz olmasıdır. Model dizisi daha karmaşıksa daha fazla veriye gerek vardır. Sürekli yoğunluklu sistemler için bu denge parametrelerin biraya getirilmesiyle sağlanır. Ayrıca HTK tamamen bağlı karışım sistemleri ve farklı olasılık sistemlerini de destekler. Bu durumlarda veri yetersizliği problemi ele alınır ve bu amaçla HSMOOTH aracı kullanılır.

Tanıma araçları

HTK da tek bir tanıma vardır, HVITE. Bu, marka geçirme (token passing)

algoritmasını kullanır. Araç izin verilebilir sözcük bölümünü tanımlayan ağı, girdi olarak alır ve SMM dizisi de girdisidir. Sözcük ağı, ses birimi ağına dönüştürülür. Daha sonra uygun SMM tanımı her bir ses birim örneğine atanır. Tanıma, saklanan konuşma dosyaları listesinde veya direkt olarak işitsel girdi üzerinde yapılır. Bu araç çapraz sözcük üçlü seslerini destekleyebilir ve çoklu hipotezler içeren kafesler (lattice) üretmek üzere çoklu tokenlarla çalışır (34).

HVITE yi çalıştırmada gerekli sözcük ağları basit sözcük döngüsüdür ya da grafiklere yöneltilir. Döngüde bigram olasılıklar normal olarak sözcük geçişlerine eklenir. Bu metin temelli formattır ve sözcük ağları direkt olarak metin editörü kullanılarak yaratılabilir. Bunu HBUILD destekler ve HLSTATS aracıda kullanılabilir.

Buna alternatif olarak daha üst düzey gramer notasyonu kullanılabilir. Bu notasyon Genişletilmiş Backus Naur (EBNF) formuna dayanır. EBNF gramer belirleme diliyle uyumludur. HPARSE bu notasyonun eşit sözcük ağına dönüştürülmesinde kullanılır.

Seçilen yöntem ne olursa olsun tanımlanan dilin örnekleri görülmelidir. Bunu HSGEN aracı başarır. Bu, ağı girdi olarak alır ve sözcük dizileri oluşturur. HSGEN ayrıca çalışmanın deneysel karmaşasını da hesaplar.

Büyük sözcüklerin yapılması kapsamında çeşitli kaynaklar birleştirilir ve her bir kaynakta dönüşüm yapılır. Bu süreci HDMAN aracı destekler.

Analiz Aracı

SMM temelli tanıyıcı oluşturulduktan sonra performansının değerlendirilmesi gereklidir. Bunu HRESULTS aracı gerçekleştirir. Bu araç konuşmacıların durmalarını, ve karışım matrislerini ve zamana göre düzenlenen transkripsiyonları sunar. Bu araç ayrıca Merit şeklinin (FOM) puanlarını ve alıcı işleme eğrisi (ROC) bilgilerini verir.

2.4. HTK İle Yapılan Çalışmalar

Konuşma tanıma sistemimizde kullanılan akustik modellerin üretimi HTK adlı araç kiti sayesinde gerçekleştirilmiştir. Bu çalışma ile ilgili ayrıntılar aşağıdaki gibidir:

2.4.1. Ses verileri

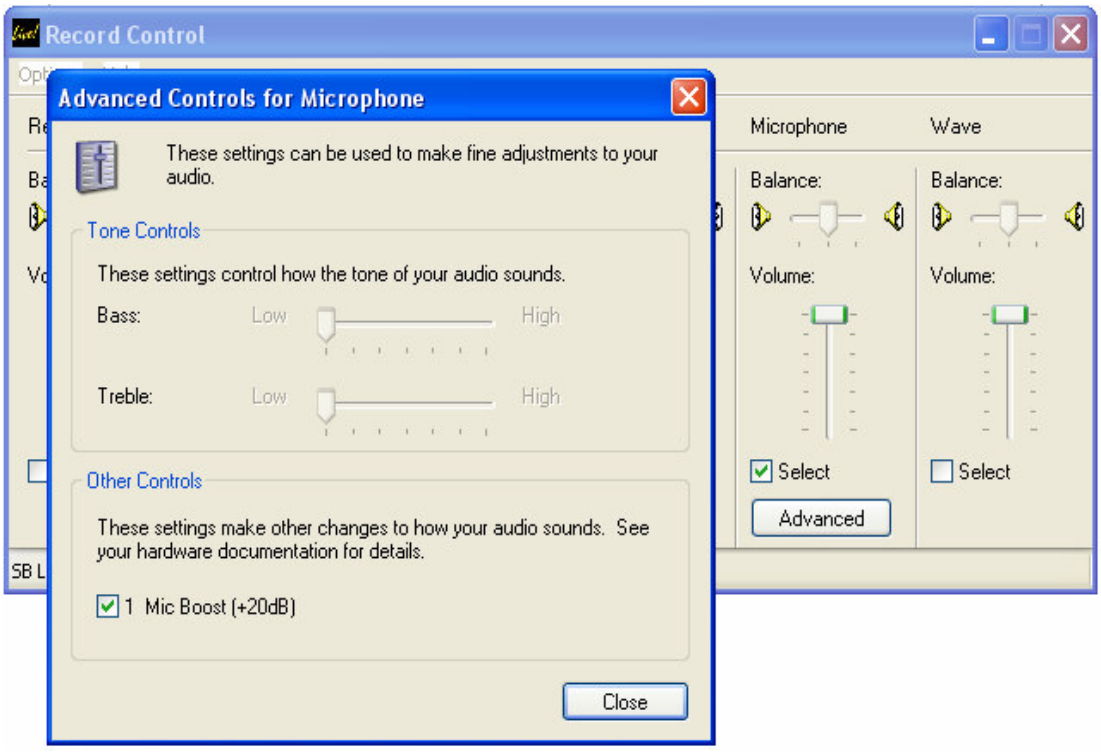
Model eğitiminde kullanılan ses verileri 16 kişiden alınan toplam 1600 adet cümleden oluşmaktadır. Her cümle ayrı bir ses dosyasına, içeriğine göre numaralandırılmış dosya isimleri ile okunmuştur. Sesler 16kHz örnekleme frekansında 16-bit mono olarak kaydedilmiştir. Cümleler sürekli konuşma olarak hazırlanmıştır. Sessizlik olan bölgeler virgül ile ayrılmıştır. Kullanılan cümlelerin seçimindeki kriter dengeli ses dağılımı ve bağlam dağılımının sağlanmasıdır.

Konuşma tanıma özelliklerinin kullanılabilmesi için sistemde bir ses kartı ve buna bağlı bir mikrofon bulunması gerekmektedir.

Kullanılan mikrofonun düşük gürültü seviyesine sahip olması aranılan özelliklerindendir. Bu sistemde düşük gürültü seviyesine sahip SENNHEISER e8455 adlı dinamik bir mikrofon kullanılmıştır.

Ses kaydetme işlemine geçmeden önce sistemdeki mikrofon ayarları yapılmıştır.

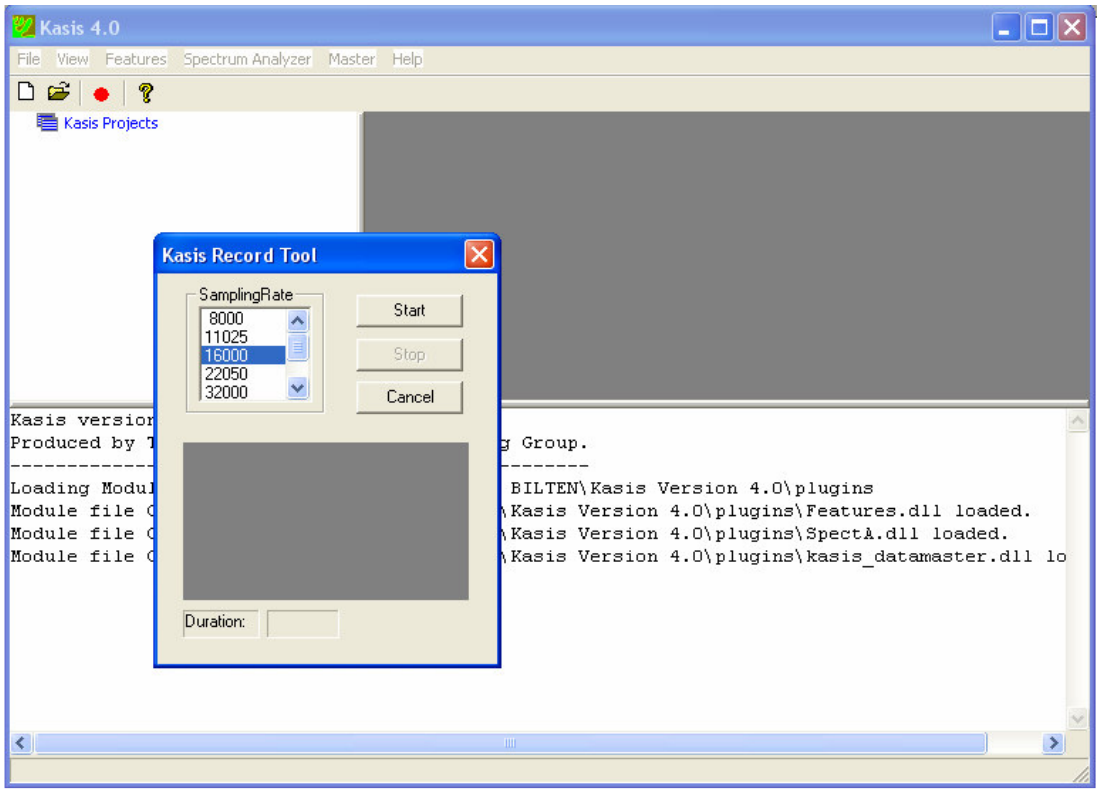
Ses ayarları kısmından mikrofon kaynağı seçili duruma getirilmiştir (Şekil 2.10).



Şekil 2.10. Bilgisayar sisteminde (Windows XP) mikrofon ayarlarının yapılması

Mikrofon ayarları tamamlandıktan sonra ses kaydetme işlemlerine geçilmiş ve bu işlem için KASİS 4.0 adında ses kaydetme programı kullanılmıştır. Ses kaydetme işlemi için Goldwave adlı bir başka ses kaydedici yazılımı kullanmak da mümkün olmaktadır.

Veri tabanında bulunan cümleler sırasıyla seslendirilmiş ve her 100 cümle bir konuşmacıya ait olacak şekilde çalışılmıştır (Şekil 2.11).

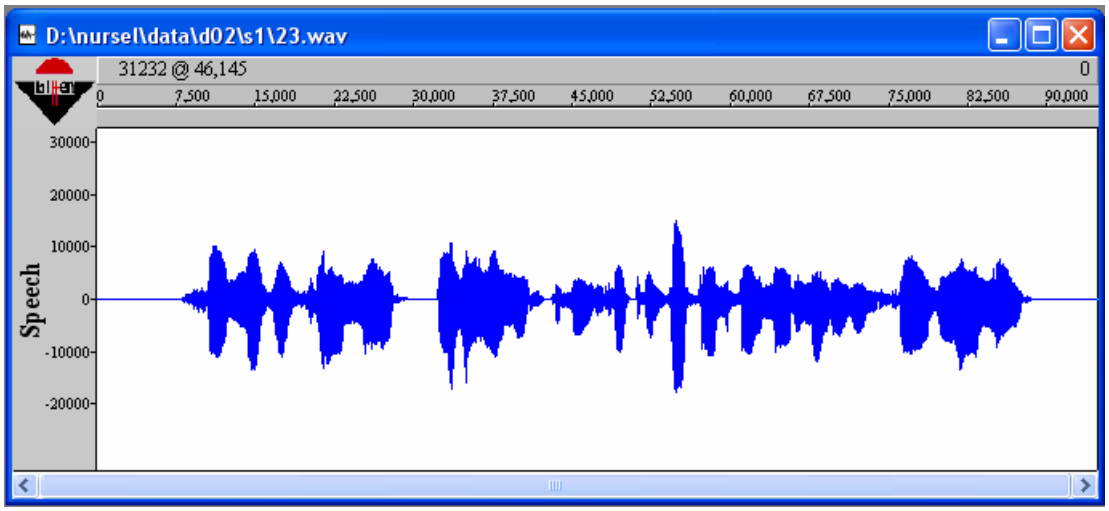


Şekil 2.11. Kasis 4.0 programında ses kaydetme işleminin başlatılma aşaması

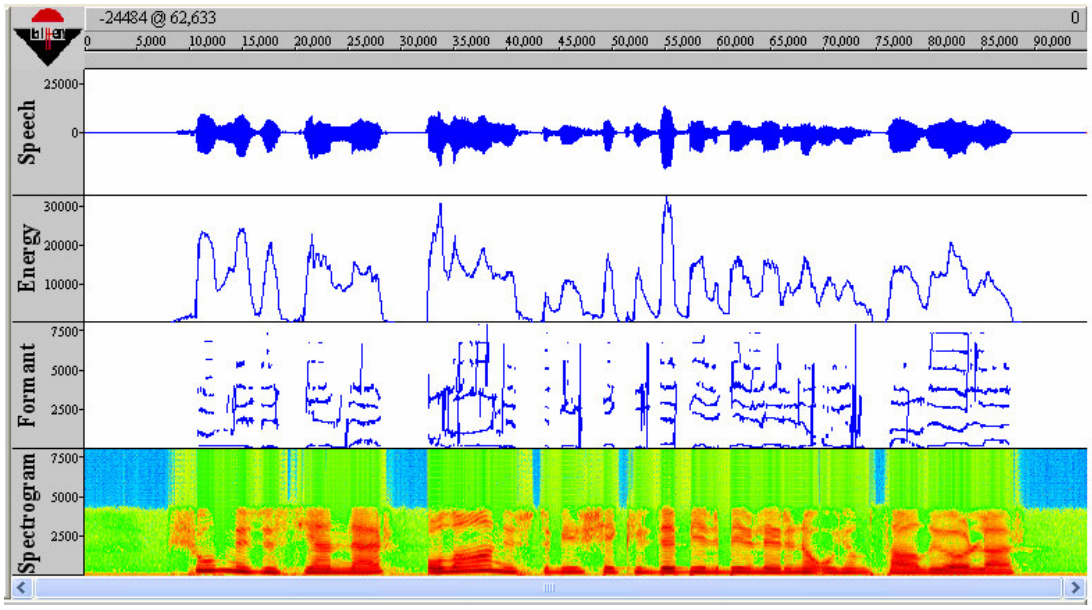
Seslendirilen cümlenin sürekli konuşma şeklinde olmasından dolayı kelimelerin başlangıç ve bitiş yerleri çok fazla belirgin değildir.

Bu da sürekli konuşma tanıma işleminin ayrık sözcük tanıma işleminden daha karmaşık bir konu olduğunu göstermektedir.

Şekil 2.12 de kaydedilen bir cümlenin ses sinyal görüntüsü görülmektedir.



Şekil 2.12. "Şu müzisyenler olağanüstü bir şekilde birbirine uyum sağlıyorlar" cümlesi



Şekil 2.13. "Şu müzisyenler olağanüstü bir şekilde birbirine uyum sağlıyor" cümlesinin spectrogramı, formant sıklığı ve enerjisi

Şekil 2.13 de bir cümlenin spectrogramı, formant sıklığı ve enerjisi görülmektedir.

2.4.2. Öznitelik vektörlerinin eldesi

Eğitimde kullanılacak verilerin eğitim sırasında birden çok sayıda defa kullanılacağı göz önüne alınarak öznitelik vektörlerinin çıkarılması işlemi eğitim öncesinde yapılmıştır. Her ses dosyası için çerçevelere (frame) ait öznitelik vektörlerini içeren *.mfc dosyaları hazırlanmıştır. Bu işlem için HTK' nın HCopy aracı aşağıdaki parametrelerle uygulanmıştır:

- Çerçeve boyutu: 25ms
- Çerçeve ölçeklendirme fonksiyonu: Hamming
- Öznitelik vektörü: MFC
- Vektör ekleri: Birinci ve ikinci türev
- Cepstrum kanal sayısı: 12
- Cepstrum yükseltme: 22

2.4.3. Teklises (monophone) eğitimi

Genel olarak geniş dağarcıklı bir sistem hedeflendiği için sözcük-altı akustik modeller kullanılmıştır. Eğitim aşamasının ilk adımı teklises (monophone) modellerinin üretilmesidir. Teklises modelleri her ses (fonem) için bir adet olacak şekilde kullanılan tüm ses dosyalarının ortalama ve varyansları kullanılarak (HCompV programı ile) ilklendirilir. Oluşturulan SMM modelleri şu özellikleri barındırır:

- Durum sayısı: 5 (sadece 3 tanesi parametre içerir)
- Topoloji: Soldan sağa. Durum atlaması mevcut değil.
- Her durum için tek karışımlı Gauss olasılık dağılım fonksiyonu.

Ardından HHEd programı ile tüm modeller aynı anda eğitime tabi tutulur. Eğitim, HTK içinde gömülü yeniden tahmin (embedded reestimation) olarak adlandırılan bir yöntem ile gerçekleştirilir. Bu yöntem Baum-Welch parametre tahmin algoritmasının tüm eğitim verileri ve modeller için paralel olarak

işletilmesini içerir. Gömülü yeniden tahmin her uygulandığında yeni bir akustik model kümesi elde edilir.

Teklises eğitimi bu şekilde 4 kere yinelemeli olarak uygulanmıştır. Akustik model listesi 29 fonemden ve konuşma olmayan bölgeleri tanımlayan “sil” (silence) modelinden oluşmaktadır. Model adları karşılık geldikleri fonem ile küçük harf olacak şekilde gösterilmiştir. Ancak ASCII kümesinde olmayan Türkçe karakterler için en yakın ASCII karakterin başına 't' harfi eklenmiştir.

Örnek

- ş → ts
- ç → tc

2.4.4. Üçlüseslere (triphone) geçiş

Üçlüsesler bağlam bağımlı teklises modelleridir. Sol ve sağ yanındaki harfe göre her sesin birden çok sayıda akustik farklılık gösteren hallerinin ayrı akustik modellerle gösterilmesi amacıyla kullanılırlar. Tanıma sırasında da sözcük içindeki (veya sözcükler arasındaki) harf sıralamalarına göre uygun üçlüses modeli Viterbi algoritmasına sokulur. Bu yaklaşımın amacı akustik farklılıkların değerlendirilerek modellerin birbirinden ayrıklaştırılmasının sağlanmasıdır.

Üçlüses modelleri ait oldukları teklises modellerinden kopyalanarak elde edilir. Eğitim verileri içinde geçen tüm üçlüseslerin listesi, HLEd programı yardımıyla bulunur. Daha sonra bu listedeki her üçlüses için HHEd programı aracılığıyla ilgili teklisesin parametreleri kopyalanır.

Üçlüses modelleri elde edildikten sonra teklises modellerinde olduğu gibi HHEd programı ile gömülü yeniden tahmin işlemi yinelemeli olarak uygulanır. Sistemde kullanılan modeller için 3 adım yineleme gerçekleştirilmiştir.

2.4.5. Karar ağacı birleştirimi (decision tree clustering)

Her fonem için sağ ve sol bağlamlarına bakarak farklı bir model oluşturmak, akustik açıdan anlamlı olsa da, istatistiksel olarak model başına düşen eğitim verisini azaltmaktadır (19). Özellikle nadir görülen üçlüsesler için yetersiz seviyede eğitim verisi bulunmaktadır. Bu nedenle Karar Ağacı Birleştirimi (KAB – DTC) algoritması kullanılmaktadır. Algoritmanın temeli üçlüseslerin ortak özelliklerinin kullanılarak bazı parametrelerinin paylaştırılması (yani ortak olarak eğitime tabi tutulması) prensibine dayanmaktadır.

QS "L_v_ince1" {"i-*","e-*"}
QS "R_v_ince1" {"*+i","*+e"}
QS "L_v_ince2" {"tu-*","to-*"}
QS "R_v_ince2" {"*+tu","*+to"}
QS "L_v_kalin1" {"ti-*","a-*"}
QS "R_v_kalin1" {"*+ti","*+a"}
QS "L_v_kalin2" {"u-*","o-*"}
QS "R_v_kalin2" {"*+u","*+o"}
QS "L_nasal" {"n-*","m-*"}
QS "R_nasal" {"*+n","*+m"}
QS "L_voicd_st" {"g-*","d-*","b-*"}
QS "R_voicd_st" {"*+g","*+d","*+b"}
QS "L_unvcd_st" {"k-*","t-*","tc-*","p-*"}
QS "R_unvcd_st" {"*+k", "*+t", "*+tc", "*+p"}
QS "L_fric_vcd" {"z-*","v-*"}
QS "R_fric_vcd" {"*+z","*+v"}
QS "L_frc_uvcd" {"s-*","ts-*","f-*"}
QS "R_frc_uvcd" {"*+s","*+ts","*+f"}
QS "L_whisper" {"h-*"}
QS "R_whisper" {"*+h"}
QS "L_oth_cons" {"y-*","r-*","l-*","c-*"}
QS "R_oth_cons" {"*+y","*+r","*+l","*+c"}

Yukarıda görüldüğü gibi sol veya sağ bağlamın ince ünlü, fısıltı, sürtünmeli ses olması gibi özellikler birleştirme kriteri olarak verilmiştir (1). Bu kriterlere göre tüm modellerin parametre içeren 2. 3. ve 4. durumları birleştirim işlemine sokulmuştur. HHEd programı içinde yer alan bu özellik sayesinde 26 bin olan tüm üçlüses modellerinin sayısı 1585' e indirilmiştir.

Bu aşamadan sonra, yeni modellerin eğitimi için 3 yinelemeli gömülü yeniden tahmin eğitimi yapılmıştır.

2.4.6. Karışım sayısının artırılması

Akustik modeller ilk aşamadan itibaren tek Gauss karışımlı olarak kullanılmaktaydı. Ancak yeterli eğitim verisi olduğunda özellikle konuşmacı bağımsız akustik modellerin elde edilebilmesi için ses farklılıkları ve değişik söylenişlerin doğru bir şekilde modellere aktarılması amacıyla birden çok sayıda Gauss karışımı kullanılması gerekmektedir.

Bu nedenle her üçlüses modeli için karışım sayısı HHEd programı kullanılarak 3' e çıkarılmıştır. Yeni oluşturulan modeller 3 yinelemeli eğitime tabi tutulmuştur.

2.4.7. Çalışma ortamı ve elde edilen dosyalar

HTK açık kaynak kodu ile dağıtılan bir araştırma kütüphanesidir. Windows platformu için uygulama programları olarak derlenmiş dosyalar web sayfasından indirilerek bilgisayara kurulmuştur. HTKBook (Hidden Markov Toolkit Book) adllı kitaptan faydalanılarak akustik model üretimi için yapılması gereken çalışmalar takip edilmiştir. İşlemler sırasında gerek olan metin dosyaları hazırlanmıştır. Bunlar ses dosyalarının listesi, metin içeriklerinin listesi, sözcüklerin hangi fonemlerden oluştuğunun listesi gibi bilgileri içeren dosyalardır. Bazı veri dosyalarının oluşturulması için Perl script dilinden faydalanılmıştır.

Akustik model üretimi bir perl dosyasının yukarıda anlatılan işlemleri sırayla yapması sayesinde gerçekleştirilmiştir. Elde edilen akustik modeller iki dosya olarak tanıma çekirdeğinin kullanımına sunulmuştur:

- hmmdefs.txt: SMM' lerin parametrelerinin tanımlandığı dosya.
- triphoneList.txt : SMM dosyalarının listesini tanımlayan dosya.

2.5. Konuşma Tanıma Çekirdeği ve Algoritmaların Tanıtımı

Konuşma tanıma sırasında en çok kullanılan yöntem Viterbi algoritmasıdır.

2.5.1. Viterbi algoritması

Tanıma için kullandığımız SMM modelleri birer sonlu durum makinesi (finite state machine) olarak görülebilir. Sonlu durum makineleri için en iyi olurluk değerini bulmaya yarayan bir optimizasyon algoritması olarak Viterbi algoritması konuşma tanıma sırasında en çok kullanılan yöntemdir (9).

Birkaç sözcükten oluşan bir dağarcığa sahip, ve her sözcük için bir SMM' nin olduğu bir konuşma tanıma senaryosu düşünelim. Elimizde tanınacak konuşma verisinden elde edilen ve SMM' lerin eğitiminde kullanılan yöntemle elde edilmiş öznitelik vektörleri dizisi olsun. Bunlara gözlem vektörleri diyeceğiz. Algoritmayı şu şekilde tanımlayabiliriz:

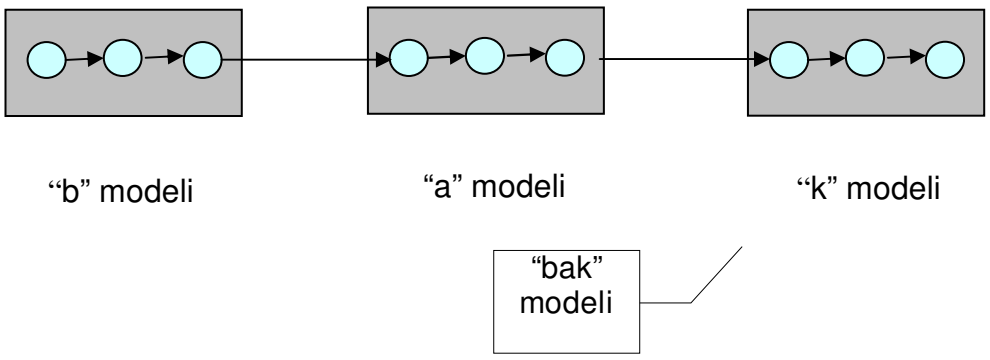
1. Her sözcük modeli için:
    2. Modelin ilk durumunu aktifleştir. Bu duruma ait skoru sıfırla.
    3. Her gözlem vektörü için (sırayla)
        4. Modeldeki tüm aktif durumlar için, eldeki gözlem vektörünün bu durum tarafından üretilme olurluğunun logaritmasını durum skoruna ekle.
        5. Her durum için durum skorunu bağlantısı bulunan tüm durumlara ilet. (Skor iletilen durumlar otomatik olarak aktifleşir)

6. Her durum için kendisine iletilen skorlardan en yükseğini, bir dahaki gözlem vektöründe kullanılmak üzere seç.
7. Son duruma ait seçilen skor SMM' nin eldeki gözlem vektörleri için vereceği en iyi olurluk değerini tutmaktadır.
8. Tüm SMM' ler arasında en yüksek genel olurluk değeri veren modeli tanınan sözcük olarak tanı.

Skor hesaplamalarında logaritma kullanılmasının nedeni olasılık değerlerinin çarpılması gerekmesidir. Kayan nokta sayılar arasında çarpım yapmak yerine logaritmalarının toplanması daha basit bir işlemdir. Düşük olurluk değerlerinde sürekli olarak çarpım yapılması, ayrıca sayı hassasiyetinin azalmasına da neden olabilecektir.

Geniş dağarcıklı tanıma

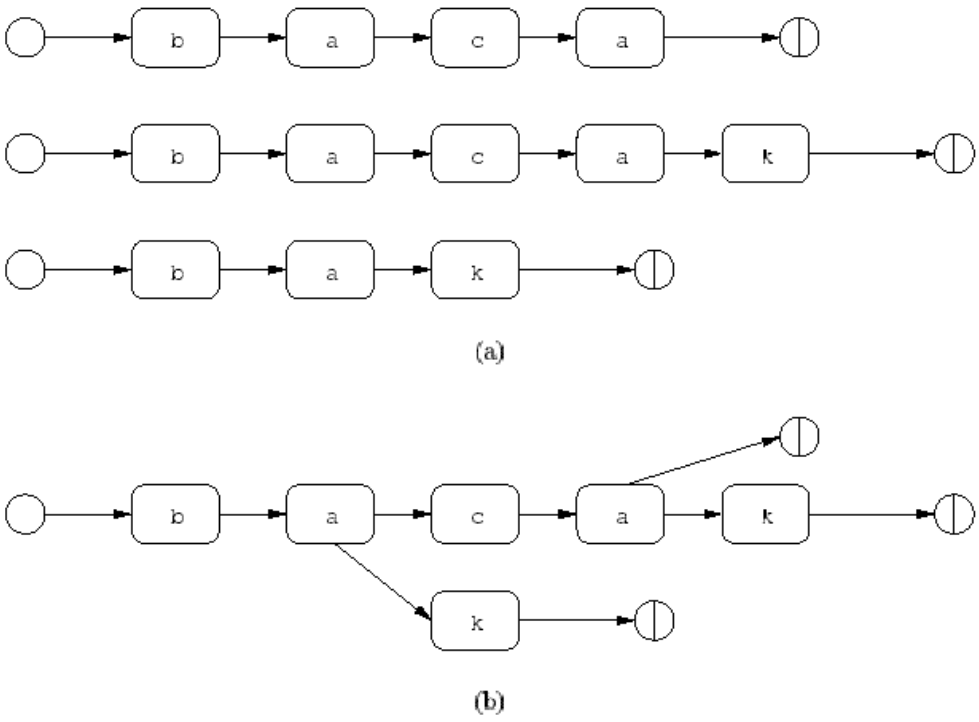
Viterbi algoritmasının tanıtıldığı yukarıdaki örnek her sözcük için bir SMM modeli olduğunu varsaymaktaydı. Bu yöntem sözcük sayısı arttığında ya da sistem dağarcığına yeni bir sözcük eklenmeye çalışıldığında sorunlar yaratmaktadır. Bu nedenle sözcük modelleri yerine fonem modelleri kullanılmıştır. Fonem modeli(fonem tabanlı sesli ifade tanıma) kullanılarak herhangi bir sözcük için bütünleşik bir SMM oluşturulması, sözcüğün içerdiği fonemlere ait SMM' lerin sırayla eklenmesiyle sağlanmaktadır (Şekil 2.14).



Şekil 2.14. SMM' leri ekleyerek oluşturulan model

Bütünleşik SMM ve tanıma ağı

Geniş dağarcıklı tanımada sözcük-altı akustik modellerin kullanımının yanısıra başka bir uygulama gereksinimi de tanınacak sözcük sayısının tanıma süresi üzerindeki olumsuz etkisinin giderilmesidir. Eğer dağarcıktaki her sözcük için ayrı ayrı bütünleşik SMM oluşturulur ve hepsi için Viterbi algoritması uygulanırsa tanıma süresi sözcük sayısıyla (uzunluklarına göre) doğru orantılı olarak artacaktır (34).



Şekil 2.15. a) Bütünleşik SMM üzerinden tanıma b) Sözcüklerin ortak harflerinden faydalanma (13)

Bu sorunun çözümü hiyerarşik olarak verimli bir yapıda ortak fonemlerin kullanılmasını sağlayan ağaç yapısıdır. Şekil 2.15 de ağaç yapısındaki bütünleşik SMM' nin ifade avantajı görülebilir: Şekilde baca, bacak ve bak sözcükleri için alternatif bütünleşik SMM' ler görülmektedir. (a) şeklindeki bütünleşik SMM üzerinden tanıma, yukarıda belirtildiği gibi verimsizdir. (b) şeklinde gösterilen ifade biçimi ise sözcüklerin ortak harflerinden

faydalanmaktadır.

Bu nedenle hem ifade daha verimli olmuştur, hem de bu bütünleşik SMM ağı üzerinden tanıma yapılırken Viterbi algoritması her gözlem vektörü için “b” modelinin durumları üzerinden sadece bir kez geçecektir. İlk şekildeki SMM için bu sayı 3' tür.

Düğümler

Şekil 2.13 (b)' de “a” ve “k” etiketi barındıran ikişer tane düğüm (node) bulunmaktadır (bkz. Şekil 2.13). Kullandığımız ağaç yapısındaki düğümler şu aşamadan itibaren birebir SMM' ler olmayacak, tanıma ağı düğümleri olarak adlandırılacaktır. Her tanıma ağı düğümü aşağıda gösterilen bilgileri taşır:

- Düğümün bir sözcük sonu olup olmadığı, öyle ise sözcüğün ne olduğu.
- Düğüme ait harf / fonem (Üçlüses adı: örnek: b-a+k)
- Düğüme ait SMM için bağlantı.
- Düğüme ait olurluk skoru.
- Varsa düğümün eş (peer) düğüm bağlantısı.
- Varsa düğümün çocuk (child) düğüm bağlantısı.

Tanıma ağacının oluşturulması

Tanıma ağacının oluşturulması sistem dağarcığındaki tüm sözcüklerin yukarıda anlatılan ağaç yapısı içinde yer alabilmesi için gerekli düğümlerin oluşturulmasıdır. Bu işlemin işleyişi aşağıdaki gibidir:

1. Başlangıç ve bitiş düğümlerini oluştur.
2. Bu düğümleri sessizlik SMM' siyle bağlantılandır.
3. Dağarcıktaki her sözcük / cümle (W) için:
    4. Üçlüses açılımını bul (örnek: “bak” sözcüğü için sırasıyla b+a, b-a+k, a-k üçlüsesleri kullanılır)

5. İlkini seç. üçlüses = ilk üçlüses
6. Tarama için başlangıç düğümünden başla. düğüm = BaşlangıçDüğümü
7. düğüm.Çocuğuvar (üçlüses) doğru olduğu sürece
    8. düğüm = düğüm.ÇocukDüğüm
    9. üçlüses = bir sonraki üçlüses.
10. Geriye kalan ve ağaçta bulunmayan üçlüsesler için
    11. düğüm.ÇocukDüğümEkle (üçlüses)
    12. düğüm = düğüm.ÇocukDüğüm
    13. düğüm.smm = üçlüsese ait SMM
    14. üçlüses = bir sonraki üçlüses.
15. düğüm.Sözcük = W

Bu işlem tamamlandığında tanıma algoritmasının tanıma yapabileceği ağaç yapısındaki tanıma ağı hazır olmuş olacaktır. Bu işlemin yapılmasından önce sistemde kullanılacak olan SMM' lerin yüklenmiş ve hazır olması gerekmektedir. Her düğüm, temsil ettiği üçlüsese ait modele bağlantılandırılır. HTK ile yapılan model eğitimi çalışmalarında HHEd programı ile, olası tüm üçlüsesler için akustik model üretilmesi sağlanmıştır. Böylece tanıma ağı, dağarcığa eklenebilecek herhangi bir sözcük veya cümleyi oluşturabilecek şekilde çalışabilir.

Tanıma işleminin gerçekleştirilmesi

Tanıma algoritması daha önce anlatılan Viterbi algoritmasının ağaç yapısındaki bütünleşik SMM yapısına uyarlanmış halidir. Viterbi algoritmasında bahsi geçen durum skorları her durum için marka (token) adı verilen yapılarda saklanmaktadır. Markalar algoritmanın kolay işlemesi açısından ikili diziler şeklinde saklanır ve ardışık gözlem vektörleri için sırayla kullanılır (her durum için iki adet marka bulunur, tanıma sırasında bunlardan bir tanesi o andaki skoru, diğeri bir sonraki gözlem için seçilen skoru saklar).

Kullanılan tanıma yönteminin bir diğer özelliği de tanıma süresinin kısıtlanması için ağaç budama tekniği kullanılmasıdır. Bu teknik tanıma

sırasında belirli bir gözlem vektörü için, o an bulunan en yüksek düğüm skorundan budama eşiğine göre daha düşük skor veren düğümlerin tanıma işleminden çıkarılması esasına dayanır. Tanıma oranını düşürmeden, yeterince uygun olmadığı görülen sözcük hipotezlerinin elenmesini sağlar.

Algoritma aşağıda anlatılan şekilde işler:

1. AktifDüğümListesini sıfırla
2. AktifDüğümListesine başlangıç düğümünü ekle.
3. Ağaç düğümlerinin markalarını sıfırla.
4. Sırayla tüm gözlem vektörleri (V) için:
    5. Her düğüm = elemanlar (AktifDüğümListesi) için:
        6. düğüme ait her durum (s) için (1..3):
            7. Olurluğu hesapla. olurluk = P(V | s). Gauss karışımlarından elde edilir.
            8. s.marka.olurluk += olurluk.
            9. Elde edilen markayı bir sonraki duruma (s+1) aktar.
                10. Aktarılan marka durumun (s+1) alternatif markasının skorundan iyiyse kabul edilir.
            11. Son durum ise (s = 3), bulunulan düğümün tüm çocuk düğümleri (çocuk) için:
                12. çocuk düğümü AktifDüğümListesine ekle.
                13. Durumun markasını düğümün ilk durumuna aktar (Kabul işlemi 10. maddedeki gibidir).
        14. düğüm sözcük sonu ise düğümün son durum markasını bitiş düğümüne (boşluk modeli içeren) ilet. Kabul işlemi madde 10 daki gibidir.
        15. Şu ana kadar rastlanan en yüksek olurluk değerini bul. eniyiskor = maksimum (düğüm skoru).
    16. Düğüm durumları için marka kümelerini yer değiştir.
    17. AktifDüğümListesindeki tüm düğümler için:
        18. Düğüm skoru eniyiskordan eşikdeğeri veya daha fazla düşük ise düğümü AktifDüğümListesinden çıkar.

Tanıma sırasında ağaç budaması eşik değeri olarak 250 kullanılmıştır. Bu algoritma tamamlandığında bitiş düğümündeki son duruma ait marka tanınan sözcük (veya cümlenin) bilgisini saklıyor olacaktır.

2.5.2. Kodların tanıtımı

Cekirdek sınıfı

XCekirdek sınıfı KTM konuşma tanıma sisteminin temel işlevselliğini sağlayan ana sınıftır. Bir Windows thread' i üzerinde çalışır. Sınıf ilklendirildiğinde ThreadFonksiyonu isimli fonksiyon üzerinden thread çalışır hale gelir.

Veriler

m_pXNetworks: Tanıma ağaçlarının tutulduğu veri yapısıdır. Tanıma ağacı birden çok sayıda olabilmektedir. KTM uygulamasında komut tanıma ağı (NetworkTypes::nt_Komut) ve fiş tanıma ağı (NetworkTypes::nt_Fis) olmak üzere iki adet tanıma ağı bulunmaktadır. Bu tanıma ağları ilgili dosyalardan NetworkYukle fonksiyonu ile yüklenirler. Değişken XNetwork sınıfından nesneler tutar.

m_tModelGroups: KTM konuşma tanıma motoru birden çok sayıda akustik model kümesi kullanılabilecek şekilde tasarlanmıştır. Bu değişken sistemin kullanacağı akustik model gruplarını belirler. Örneğin konuşmacı bağımlı veya konuşmacı bağımsız akustik model gruplarının kullanımı için uygundur. KTM sistemi, HTK ile elde edilen akustik modellerin başarılı tanıma performansı sayesinde tek akustik model kümesi ile yüksek tanıma oranı sağlamaktadır. Gelecek çalışmalarda, istendiği takdirde alternatif akustik model kümeleri kullanıma açılabilecektir.

m_eSeciliModelGroup: Model grupları arasından, anlık olarak aktif / seçili olan model grubunu belirler. Sistemin varolan tek model gruplu hali ile

mg_Native değerini taşıması beklenir.

m_eSeciliNetworkType: Tanıma ağları arasından, o anda seçili olan tanıma ağını gösterir. Fiş tanıma yapılırken nt_Fis, komut tanıma yapılırken nt_Komut değerini taşıması beklenir.

s_szNetworkDosyaAdlari: Tanıma ağları için dosya adları listesi. Tanıma ağları yüklenirken bu sabit değişkendeki dosyalar üzerinden ağ içerikleri okunur.

s_szModelDosyaAdlari, s_szModelListDosyaAdlari: Her model grubu için bir tane SMM tanım dosyası ve SMM model listesi dosyası gerekmektedir. Bu değişkenler her model grubu için gerekli dosya adlarını tutmaktadır.

Diğer değişkenler Xcekirdek sınıfının thread üzerinden çalışmasını yönetmek için gereken verileri saklar. Thread numarası, müşteri thread ve Xcekirdek thread'i arasında işlem senkronizasyonunu sağlayacak olay (event) nesneleri, kullanıcı arayüzüne mesaj gönderilmesini sağlayan pencere numarası (HWND) gibi değişkenlerdir.

2.5.3. Yöntemler (sınıf metodları – member functions)

Ext_ModelGrubuBelirle: XCekirdek thread fonksiyonuna model grubu değişikliği yapılması istendiğini bildirir. Thread'in mesaj kuyruğuna komut ile ilgili mesajin yazılmasını sağlar. Kullanıcı arabirimi kodu tarafından çağırılır.

Ext_SesKayitOlayi: Kullanıcı arabirimi kodu tarafından, konuşma tanıma istemi ile ilgili olarak çağırılır. İlk parametresi hangi tanıma ağının kullanılacağını (fiş, komut), ikinci parametresi ise ses kaydının başlması veya bitmesi gerektiğine dair emri saklar. Thread'in mesaj kuyruğuna ilgili komutun eklenmesini sağlar.

ThreadFonksiyonu: XCekirdek sınıfının en önemli fonksiyonudur.

İlklendirmeden sonra sürekli olarak çalışır ve komut bekler. İlklendirme olarak şu işlemler yapılır:

- Kullanıcı arabirimi thread'ine başlandığı sinyalini gönder.
- Win32 ses kayıt ilklendirmesini yap.
- Döngü değişkenlerini ilklendir.

Çalışma döngüsü sırasında karşılaşılabilecek komutlar şunlardır:

- Ses kayıt tamponu doldu: Eğer sistemin saklayabildiğinden uzun bir ses kaydı yapılmakta ise, kayıt işlemini durdurur ve kullanıcıyı bilgilendirir.
- Çıkma mesajı (WM_QUIT) geldi: Thread' in çalışması sonlandırılır.
- Ses kaydı başlatma mesajı geldi: Kayıt işleminin başladığı yerel değişkende not edilir. Win32 ses kaydı başlatılır. Tanımanın yapılacağı tanıma ağı seçimi sınıf üye değişkenine işaretlenir.
- Ses kaydı sonlandırma komutu geldi: Ses kayıt işlemi sonlandırılır, bu durum yerel değişkene not edilir. Kaydedilen sesler birleştirilerek SoylenisiTanı üye fonksiyonu çağırılır.
- Model grubu değiştirme komutu geldi: ModelGrubuBelirle üye fonksiyonu, parametre olarak verilen model grubu seçimi ile çağırılır.

HatamesajiGonder: Fonksiyona girdi olarak verilen değişken argümanlı (vararg) hata mesajı bir tampona aktarılarak kullanıcı arabirimi penceresine mesaj olarak gönderilir. Tamponun bellekten temizlenmesi kullanıcı arabirimi penceresi tarafından gerçekleştirilir.

ModelGrubuBelirle: Parametre olarak verilen seçili olan model grubunu değiştirir ve ModelGroupYukle fonksiyonunu çağırır.

ModelGroupYukle: Tanıma ağ -lar- ında bulunan SMM bağlantıları model grubundaki SMM nesnelerine bağlanmaktadır. Bu nesneler tanıma sırasında Viterbi algoritması tarafından kullanılacak *durum*ları saklayan nesneleri içerir.

Bir akustik model grubunu tanımlayan nesneler şunlardır:

- SMM nesnelerinin saklandığı model kümesi (XModelSet).
- SMM' ler arasında ortak olarak kullanılan durum nesnelerinin saklandığı durum kümesi (XStateSet).
- SMM' ler arasında ortak olarak kullanılan ve durumlar arası geçiş olasılıklarını saklayan geçiş olasılık matrisleri kümesi (XTransSet)
- SMM modellerinin adlarını saklayan model adları sınıfı (XModelList).

Fonksiyon işlemine başladığında öncelikle, eğer varsa bu nesnelerin yüklenmiş olanlarını siler. Sonrasında tanımlı olan tüm model grupları için aşağıdaki işlemleri yapar:

- SMM tanımlama dosyasını açar
- Bu dosyada SMM' ler tarafından ortak kullanılan: durumlar arası geçiş matrisleri (transition matrix), durum verileri (state) ve SMM tanımları bulunur.
- ModelGrubuna ait m_pxModelSet, m_pxStateSet ve m_pxTransSet üye değişkenleri ilklendirilir.
- Dosya içinden bu nesnelere ait veriler sırayla okunarak ilgili üye değişkenin nesne havuzuna eklenir.
- Dosya okunduktan sonra model adlarını içeren model listesi dosyası okunarak model grubu içindeki m_pxModelList değişkeni içine yerleştirilir.

NetworkYukle: Tüm tanıma ağları için aşağıdaki işlemler yapılır:

- Tanıma ağını tanımlayan dosya adı s_szNetworkDosyaAdlari değişkeninden okunarak dosya açılır.
- m_pxNetworks değişkeni üzerinden CreateNetwork fonksiyonu ile tanıma ağı dosyadan yüklenerek düğümler ve SMM bağlantıları oluşturulur.

SoylenisiTani: Ses kaydı tamamlandıktan sonra çağırılır. Ses kaydı

fonksiyonları tarafından kaydedilen ses tamponları birleştirilerek bir ses dosyası halinde saklanmış ve MFC öznitelik vektörleri çıkarılarak bir dosyaya yazılmış durumdadır. XObservations sınıfından bir nesne yaratılarak MFC dosyasındaki gözlem vektörleri okunur. Bu nesne sonra seçili olan tanıma ağının RecognizeUtterance fonksiyonuna gönderilir.

Tanıma işlemi sonucunda bir tamsayı dönen fonksiyonun çıktısı kullanıcı arabirimi penceresine mesaj olarak gönderilir. Gönderilen tamsayı tanınan sözcük / cümlenin seçili olan tanıma ağındaki sıralamasıdır.

2.5.4. Çalışma – kullanılma şekli

XCekirdek sınıfının kullanımı şu şekilde gerçekleştirilir:

- Kullanıcı arayüzü tarafından bir XCekirdek nesnesi yaratılır ve ilklendirilir.
- Akustik model gurubu belirlemek için Ext_ModelGrubuBelirle fonksiyonu çağırılır.
- Tanıma amaçlı ses kaydının başlaması için Ext_SesKayitOlayi fonksiyonu çağırılır.
- Ses kaydının tamamlanması için fonksiyonu Ext_SesKayitOlayi yeniden (farklı parametrelerle) çağırılır. Bu fonksiyonun çağırılmasından bir süre sonra nesnenin ilklendirilmesi sırasında parametre olarak verilen pencere için konuşma tanıma sonucu bildiren mesaj alınır. Bu mesajdaki değerlere göre komut tanıma veya fiş tanıma sonucu arabirimde görselleştirilir.

XNetwork Sınıfı

Tanıma ağları bu sınıf üzerinden işlev gösterir. Tanıma ağı XNode türünde nesnelerden oluşan bir ağaç yapısı şeklinde oluşturulur. Ağacın oluşturulması, tanıma ağının içeriğini belirleyen dağarcık dosyasının okunması ile gerçekleştirilir.

Veriler

m_pRootNode: XNode (düğüm) türünden nesne. Tanıma ağındaki ilk düğüm. Dağarcıktaki tüm sözcük ve cümleler için başlama noktası. Boşluk (sil) SMM' sine bağlantılandırılmıştır.

m_pEndNode: Bitiş düğümü. Boşluk SMM' siyle bağlantılandırılmıştır. Tanıma ağacındaki tüm sözcük ve cümleler bu düğüme bağlanırlar.

m_pModelSet: SMM'lerin saklandığı veri yapısıdır. XModel sınıfından nesneler tutar. Model listesi yüklenmesi XCekirdek tarafından gerçekleştirilir.

m_pModelList: SMM isimlerinin ve listesinin saklandığı veri yapısıdır. yüklenmesi XCekirdek tarafından gerçekleştirilir.

m_pObservations: Tanınacak konuşmaya ait gözlem vektörlerini saklayan veri yapısıdır. Oluşturulması XCekirdek tarafından gerçekleştirilir.

m_pWordList: Tanıma ağacı dağarcığında bulunan tüm sözlerin sırayla yer aldığı bir dizidir (array).

m_iNumNodes: Tanıma ağacında bulunan toplam düğüm (XNode) sayısını tutar. Tanıma ağacının yüklenmesi sırasında değeri belirlenir. Bu değere başlangıç ve bitiş düğümleri dahil değildir.

m_iNumVocabulary: Tanıma ağacı dağarcığında bulunan toplam söz (sözcük / cümle) sayısını saklar. Değeri tanıma ağacının yüklenmesi sırasında (CreateNetwork fonksiyonu) hesaplanır.

2.5.5. Sınıf Metodları

CreateNetwork: Dağarcık dosyasını kullanarak tanıma ağını ve düğümleri oluşturan fonksiyondur. Model kümesi ve model listesini XCekirdek' ten alır.

Öncelikle başlangıç ve bitiş düğümlerini oluşturur. Ardından dağarcık dosyasını açarak içindeki geçerli (a-z, C, G, I, O, S, U dışında karakter barındırmayan) *söz*ler için gerekli düğümleri oluşturarak tanıma ağı içine, ağaç yapısını sağlayacak şekilde ekler (Yardımcı fonksiyon AddWordPath). Burada işleyen algoritma *Tanıma Ağacının Oluşturulması* başlığı altında anlatılmıştır. Dağarcık dosyasındaki tüm sözler aynı şekilde işlenerek ağaç yapısı tamamlanır.

RecognizeUtterance: XCekirdek tarafından, kaydedilen söylenişin tanınması için çağırılan fonksiyondur. Parametre olarak söylenişe ait gözlem vektörleri içeren XObservations nesnesi alır. Çalışma şekli şöyledir:

- Tanıma değişkenleri (örnek XNode nesneleri için marka değişkenleri, aktif düğüm listesi, anlık eniyi düğüm skoru vs.) sıfırlanır.
- Gözlem vektörleri içindeki her vektör için:
  - Viterbi algoritmasını aktif düğümler üzerinde uygulayan ProcessObservation fonksiyonu çağırılır.
  - Düğümlerin ikili marka seti bir daha ki gözlem vektörü için yer değiştirilir.

ProcessObservation: Aktif olan tüm düğümler için şu işlemleri yapar:

- Seçili gözlem vektörünü düğüme göndererek olurluk değerlerinin hesaplanmasını ve markalara kaydedilmesini sağlayan XNode::ProcessObservation fonksiyonunu çağırır.
- Düğümün birdahaki gözlem vektörü için aktif olup olmayacağını hesaplar. Budama işlemini kontrol eden bu fonksiyon XNode::SurvivesPruning ile gerçekleştirilir.

XNode Sınıfı

Tanıma ağını oluşturan birimler XNode sınıfından yaratılmış nesnelerdir. XNode hem SMM bilgisini hem de tanıma ağı hiyerarşisini saklayan ve aynı

zamanda Viterbi algoritmasını barındıran bir veri yapısı olarak tanıma işleminin en önemli unsurlarından biridir. Uygulanmakta olan tanıma algoritması marka geçirme (token passing) yöntemi ile çalışır be budamaya izin verecek şekilde tasarlanmıştır.

Veriler

m_pPeerCreation: Tüm düğümlere doğrusal olarak erişmeyi sağlayan bağlı liste (linked list) için değişken. Bu düğümden sonra yaratılan düğüme bağlantı kurmaya yarar. Tüm düğümler için yapılması gereken işlemlerde Creation serisi değişkenler kullanılır.

m_pNextNetwork: Ağaç yapısı içinde bu düğümün altında olan çocuk düğüm. Bağlı liste (linked – list) ile oluşturulan ağaç için child bağlantı.

m_pPeerNetwork: Ağaç yapısı içinde bu düğüm ile aynı seviyede olan, kardeş düğüm. Bağlı liste (linked – list) ile oluşturulan ağaç için peer bağlantı.

m_pPeerActive: Aktif düğümlerin saklandığı bağlı liste için değişken. Bu düğümden sonraki aktif düğümün adresini saklar. Bu değişken gözlem vektörleri boyunca sıra ile kullanılan ikili şekilde tasarlanmıştır.

m_iTouchActive: Düğümün aktif düğüm listesinde en son ne zaman güncellendiğini saklayan değişken.

m_pTokens: Düğümün sahip olduğu SMM' nin her düğümü için iki adet marka bulunur. Markalar Viterbi algoritması sırasında ortaya çıkan olurluk değerlerini (skor) ve eni iyi durum dağılımını saklayan verilerdir. İki marka gözlem vektörleri değiştikçe sırayla kullanılır.

m_pModel: Düğüme ait akustik modeli gösteren değişken. Birden çok sayıda düğüm aynı SMM' ye işaret ediyor olabilir.

m_fNodeScore: Düğüme ait skor. Budama işlemi sırasında kullanılır..

m_pWordUtterance: Eğer düğüm dağarcıktaki bir sözün son harfine aitse bu değişken dağarcıktaki ilgili sözü gösterir.

Statik Değişkenler

s_iDualSelector: m_pPeerActive gibi ikili değişkenler için hangisinin kullanılacağını gösteren endeks değişkeni.

s_iClock: Touch değişkenleri için saat. Her gözlem vektörü için artırılır.

s_pActiveFirst: Aktif düğümler bağlı listesinin ilk elemanı.

s_pCreationFirst: Düğüm bağlı listesinin ilk elemanı.

s_iNumActive: Aktif düğüm sayısı.

s_fBestNodeScore: Her gözlem vektörü işlenmesi sonucunda, elde edilen en yüksek değer. Bu değer ağaç budama işlemi sırasında kullanılır.

s_fPruningThreshold: Budama eşiği. Olurluk değeri en yüksek düğüm değerinin eşik veya daha altında kalan düğümler budanır ve atif düğüm listesinde yer almaları engellenmiş olur.

s_pObservationVector: Gözlem vektörünü gösterir. Xnetwork tarafından belirlenir.

s_tFinalToken: Tanıma sonucunun ulaştığı marka (token) verisi. En yüksek olurluğa sahip hipotez bu değişkenden okunur.

Sınıf Metodları

AddChild: Ağaç yapısı içinde çocuk düğümler yeni düğüm ekler.

FindChild: Ağaç yapısı içinde bu düğümün altında yer alan düğümler arasında (ilk seviyede) parametre olarak verilen üçlüsese ait bir düğüm varsa onun adresini döner. Düğüm bulunamadı ise sıfır (boş) döner.

ResetNode: Her tanıma işlemi başlamadan önce statik olan ResetRecognition fonksiyonu tarafından çağırılır. Düğümün tanıma işleminden önce sıfırlanması gereken değişkenlerini düzenler.

UpdateToken: Verilen marka (token) seçilen duruma uygunsa (skoru daha yüksek ise) durumun markasını günceller. Yapılan güncelleme bir daha ki gözlem vektörü için geçerli olacak şekilde yapılmaktadır. Düğümü (eğer değilse) aktif düğümler listesine ekler.

UpdateTokenExternal: Tanıma ağacının üst tarafından iletilen marka için düğüm içinde güncelleme yapar. Bkz. UpdateToken.

UpdateFinalToken: s_tFinalToken değişkenini günceller.

ProcessObservation: Viterbi algoritmasının düğüm üzerinde gerçekleştirilen kısmıdır. Tüm durumlar (her durum için ikili marka değişkenleri vardır) için aşağıdaki işlemler gerçekleştirilir:

- Eğer durum aktif değilse bir sonraki duruma geçilir.
- Duruma ait markanın skoru, düğümün SMM' sindeki ilgili durum (XState) tarafından o anda kullanılmakta olan gözlem vektörü için üretilen olurluk değeri eklenerek güncellenir.
- Yeni marka PassToken fonksiyonu ile bağlı düğümlere iletilir.
- PassToken: Viterbi algoritmasının gerçekleştirilmesindeki önemli bir işlem

olan marka iletimi (token passing) bu fonksiyon içinde yürütülür. Kaynak durumdan diğer durumlara marka iletimi amacıyla tüm durumlar için şu işlemler yapılır:

- Kaynak durumdan hedef duruma geçiş olasılığı bulunur (XTransMatrix).
- Geçiş olurluğu sıfıra çok yakın ise bir sonraki duruma geçilir.
- İletilecek markaya geçiş olasılığının logaritması eklenir.
- Hedef durum SMM' ye ait bir durumsa UpdateToken çağırılır.
- Hedef durum son durum ise m_pNextNode ile başlayan tüm düğümler için UpdateTokenExternal çağırılır.

Statik Değişkenler

Static_CreateNode: Yeni bir düğüm yaratır ve bağlı düğüm listesi değişkenlerini günceller.

Static_DestructNodeList: Bağlı düğüm listesini kullanarak tüm düğümleri siler.

Static_SetNetworkStartEnd: Statik tanıma ağı başlangıç ve bitiş düğümlerini belirler.

Static_ResetRecognition: Her tanıma işlemi öncesinde XNode sınıfının statik değişkenlerinin sıfırlanmasını sağlar. Aktif düğüm listesi, eniyi skor, saat değişkeni gibi değişkenleri ilk değerlerine getirir.

Static_PostObservationJobs: Her gözlem vektörünün işlenmesinden sonra çağırılır. İkili değişkenlerin yer değiştirmesini sağlar. Aktif düğüm listesini sıfırlar.

2.5.6. Akustik Modellerle İlgili Sınıflar

XNetwork ve XNode sınıfları tanıma ağacının hiyerarşik yapısını tanımlar ve

ağaç dizilimli ve ışın budama (beam pruning) yapılabilen bir tanıma algoritması için gerekli verileri ve algoritmaları içerir. Akustik modellemeyle ilgili kısımlar ise XModel, XState, XTransMatrix sınıfları aracılığıyla uygulanmıştır. Bu sınıfların hepsi bağlı listelere izin veren XSetItem sınıfından türetilmişlerdir ve sırasıyla XModelSet, XStateSet, XTransMatrixSet adlı sınıflar aracılığıyla kümeler halinde bellekte yerleştirilirler. Bahsedilen tüm sınıflar HTK tarafından üretilen MMF (Master Model File) türü SMM dosyalarını okumak üzere hazırlanmışlardır.

XModel

Bir SMM ile ilgili bilgileri tutan sınıftır.

Veriler

m_szName: Akustik modelin adı. Modelin karşılık geldiği üçlüses buraya yazılır.

m_iNumStates: Modelin kaç durumdan (state) oluştuğunu saklar.

m_pStates: Modele ait durumlar bu dizi değişkeni içinde gösterilir. Durumlar modeller arasında paylaşılırlar, bu yüzden bir durumu birden çok sayıda model kullanabilir.

m_pTransMatrix: Durumlar arası geçiş için olasılık dağılım matrisi. Durumlar gibi geçiş matrisleri de ortak kullanılabilirler.

Sınıf Metodları

LoadFromFile: Bir XModel nesnesini SMM tanım dosyasından okuyarak durum ve olasılık dağılım bilgilerini belirleyen fonksiyondur. Önce durum sayısını okur ve durumları durum listesinden bulur veya listeye yeni durum ekler. Daha sonra geçiş olasılık dağılım matrisi için benzer işlemleri

gerçekleştirir.

XState

SMM için durum bilgilerini saklayan sınıftır. Durumlar gözlem vektörü olurluk değerlerinin hesaplanması için olasılık dağılım fonksiyonları parametreleri barındırırlar. Bu çalışmada kullandığımız durumlar sürekli yoğunluklu Gauss karışımlarına yer vermektedir.

Veriler

m_szName: Durumun adıdır. Paylaştırılmış durumlar “e_s320” gibi adlar alırlar.

m_iNumMixes: Durumun Gauss karışımı sayısıdır.

m_iTouchOutputProb: Her gözlem vektörü için olasılık değerinin sadece bir kere hesaplanmasını sağlayan bayrak değişkeni.

m_pMeanVectors: Gauss karışımları için ortalama vektörleri.

m_pVarianceVectors: Gauss karışımları için varyans vektörleri.

m_fMixtureWeights: Her karışımın olurluk değerine etki oranı (0..1)

m_fLastOutputProb: Bir gözlem vektörü boyunca geçerli olan olurluk değeri.

Sınıf Metodları

LoadFromFile: Durumu SMM tanım dosyasından okuyan fonksiyon.

PassToLogSpace: Karışım ağırlıklarının logaritmasını alır. Yüklemeden sonra bir kere çağırılır.

ResetState: Her söyleniş tanıma işleminden önce bir kere çağırılır. Bayrak değişkenini sıfırlar.

GetOutputProb: Parametre olarak verilen gözlem vektörü için olurluk değeri hesaplar ve m_fLastOuputProb değişkenine not eder. Aynı gözlem vektörü için bir daha çağırılması durumunda hesaplanmadan not edilmiş değer döndürülür.

XTransMatrix

Durumlar arası geçiş olurluk değerlerini saklayan veri yapısıdır. Hem Viterbi' de olurluk hesaplanmasında, hem de bir durumdan hangi durumlara geçiş yapılabileceğinin belirlenmesinde kullanılır.

Veriler

m_szName: Matrisin adını tutar. Örnek “T_v”

m_pTransitionProbs: Olasılık geçiş değerleri.

m_iMatrixSize: Kare matrisin boyutu.

Sınıf Metodları

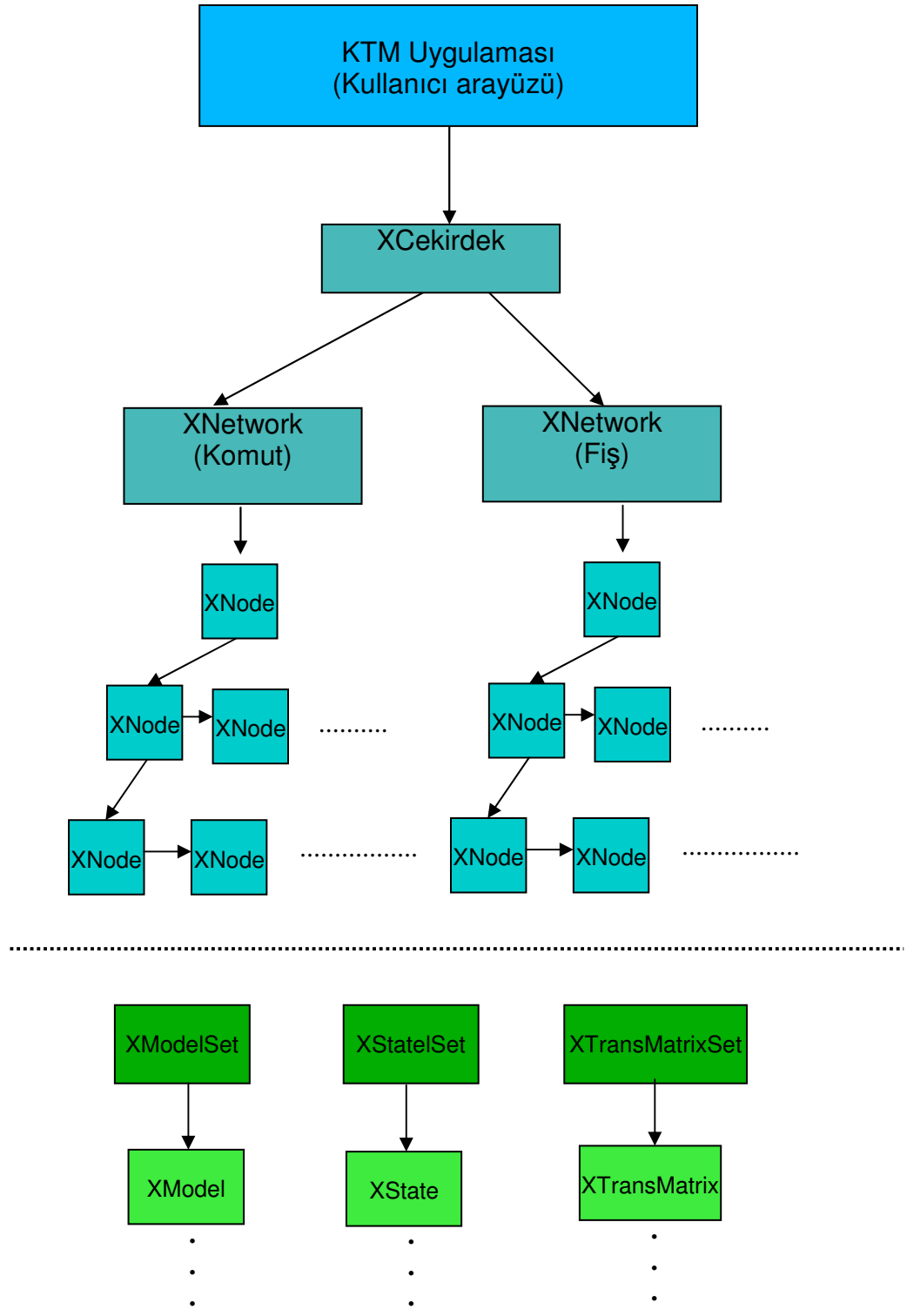
LoadFromFile: Matrisi dosyadan yükler.

PassToLogSpace: Matris elemanlarının logaritmasını alır. Bir kere çağırılır.

GetTransProb: Verilen kaynak durumdan hedef duruma geçiş olurluğuna döner.

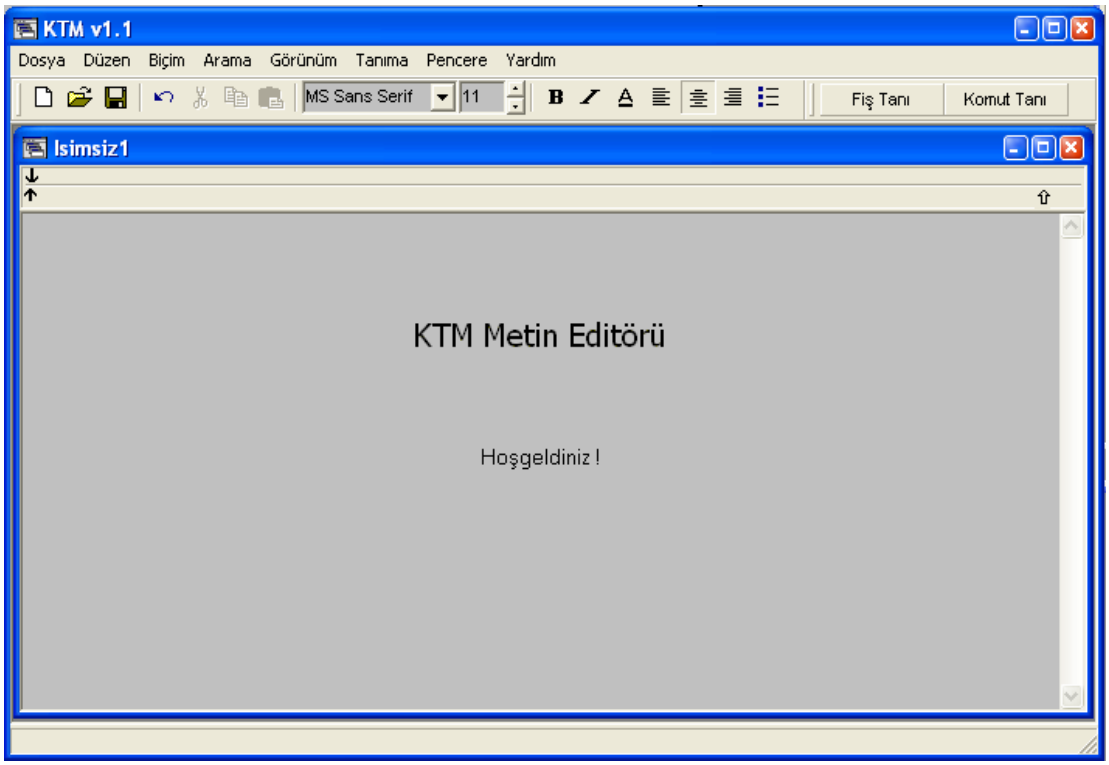
3. KTM YAZILIMININ GELİŞTİRİLMESİ ve KULLANIMI

Geliştirilen konuşma tanıma sisteminin tasarımı Şekil 3.1 de görülmektedir.



Şekil 3.1. KTM Uygulaması kullanıcı arayüzü tasarımı

KTM, konuşma tanıma sistemi entegre edilmiş bir metin editörüdür. Arayüzü Şekil 3.2 de görüldüğü gibidir. Kullanım özelliklerine ait bilgiler Yardım dosyası içinde yer almaktadır.



Şekil 3.2. KTM metin editörü

KTM Borland C++ Builder programlama ortamı içinde C++ dili ile geliştirilmiştir. Rich Text Format adı verilen dosya formatında çalışmak üzere tasarlanmıştır. Bu formatta font, paragraf renk vs. özellikleri bulunan metin elemanları istenen sayfa düzeni içinde yerleştirilebilmektedir.

3.1. Konuşma Tanıma Entegrasyonu

KTM uygulaması konuşma tanıma işlevlerini sağlamak amacıyla XCekirdek sınıfından bir nesne yaratır. Çalışma süresi boyunca bu nesne ile KTM uygulaması arasında çift yönlü işlem / bilgi akışı olur.

3.2. KTM' den XCekirdek' e

- Uygulamanın başlamasının hemen ardından XCekirdek nesnesi yaratılır.
- Nesneye ait Ext_ModelGrubuBelirle fonksiyonu çağırılır. Bu akustik modellerin yüklenmesini sağlar. Birden çok sayıda akustik model seti ile çalışılabilecek şekilde tasarlanan programda genel kullanıma uygun tek model seti uygulama halindedir.
- Fiş Listesi kullanıcı tarafından değiştirilirse Ext_ModelGrubuBelirle fonksiyonu yeniden çağırılır.
- Komut tanıma veya Fiş Tanıma düğmesine basıldığında Ext_SesKayitOlayi fonksiyonu kayıt başlangıcı ibaresi ile çağırılır. Fonksiyon parametresi olarak hangi düğmenin basıldığı gönderilir. Bu fonksiyon çağırıldığında ses kaydı işlemi başlar.
- Komut tanıma veya Fiş Tanıma düğmesi bırakıldığında Ext_SesKayitOlayi fonksiyonu kayıt bitişi ibaresi ile çağırılır. Ses kaydı sonlandırılır ve tanıma işlemi başlatılır.

3.3. XCekirdek' ten KTM' ye

Sınıfın ilklendirilmesi sırasında KTM ana penceresine ait pencere kodu (windows handle, HWND) XCekirdek tarafından bir değişkene kaydedilir. Asenkron olarak oluşan tüm olaylarda bu kod aracığı ile Windows mesajı gönderilerek haberleşme sağlanır.

- XCekirdek ilklendirilmesi tamamlandığında ana pencereye WM_CORE_READY mesajı gönderilerek çekirdeğin konuşma tanıma yapmaya hazır olduğu bildirilir.
- Tanıma işlemi sonuçlandığında tanınan içerik WM_RECOGNISEDTEXT mesajı aracılığıyla KTM anapenceresine gönderilir. Mesaj parametreleri olarak Komut / fiş sonucu olduğu, ve ilgili dağarcık içinde tanınan komut / fişin numarası iletilir.
- Herhangi bir durumu kullanıcıya bildirmek amacıyla

WM_SHOWMESSAGE aracılığıyla bir katar (string) gönderilebilir.

Anapencere bu mesajı aldığında gönderilen katarı bir mesaj kutusu içinde gösterir.

3.3.1. Thread yapısı

KTM uygulaması her Win32 uygulamasında olduğu gibi bir kullanıcı arayüzü thread' i üzerinde çalışır. Kullanıcı arayüzü thread' leri olay yönelimli olarak işlev gösterir. Böylece sadece kullanıcıdan gelen komutlar veya başka kaynaklardan gönderilen sinyaller üzerine aktivite gösterir. KTM uygulaması için bu tip sinyallerden biri de XCekirdek tarafından gönderilen mesajlardır. Yukarıda anlatılan mesajlardan tanıma mesajı geldiğinde tanınan içeriğin komut veya fiş olmasına göre aşağıdaki işlemler gerçekleştirilir.

- Komut: Komutun karşılık geldiği menü seçeneği işletilir.
- Fiş: Tanınan metin seçili metin penceresi içine yazdırılır.

KTM uygulamasındaki ana thread' in aksine XCekirdek tarafından yaratılan thread bir hizmet thread' idir. Bir işlem emri geldiğinde (Ext_xxx fonksiyonları aracılığıyla) verilen emirdeki işlem tamamlanana kadar çalışmasını sürdürür. Örneğin bir tanıma işlemi başladığında seçilen tanıma ağı üzerinden önceki bölümlerde anlatılan Viterbi algoritması işlemeye başlanır ve tanıma tamamlanana kadar sürdürülür.

Bu tasarım bir tür çok-görevli olarak düşünülmelidir. Paralel olarak çalışan thread' ler aralarında Win32 API' sinin sağladığı PostMessage ve PosteMessageToThread fonksiyonları aracılığıyla iletişim sağlanır.

3.4. Yazılım Modüllerinin Tanıtımı

KTM bir MDI (Multiple Document Interface – Çoklu Dokuman Arayüzü)

yazılımıdır. Bu yazılım türü tasarımında dökümanlara bağlı görünüm (view) pencereleri – formları ve bunlara ev sahipliği yapan ana pencere - form bulunur. Ana pencere ve editör penceresi ile ilgili temel özellikler şunladır:

3.4.1. Editör penceresi

TEditorPenceresi sınıfı ile uygulanmıştır.

- Cetvel nesnesi ile ilgili düzenlemeleri gerçekleştirilir (RulerItemMouseDown, RulerResize, RulerItemMouseMove, FirstIndMouseUp, RightIndMouseUp, LeftIndMouseUp).
- Pencerenin boyut değişimleri ve yeniden çizimi (FormResize, FormPaint).

Bunun dışındaki temel işlevler taban sınıfı TForm tarafından gerçekleştirilir.

3.4.2. Ana pencere

TAnaPencere sınıfı ile uygulanmıştır.

- Yeni metin penceresi açma (FileNew1Execute).
- Yeni dosya açma (FileOpen1Execute)
- Dosyayı kadetme (FileSave1Execute)
- Tanımabaşlatma işlemleri (btnSesTaniMouseUp, btnSesTaniMouseDown)
- Araç çubuğu işlemleri (miChangeToolbarsClick, , miTanimaSelectionClick)
- Yazıtipi işlemleri (cbFontNameChange, edtFontSizeChange, miYaziTipiClick)
- Fiş listesi düzenleme (miFisListesiClick).
- Büyük / küçük harf düzenleme (miBuyukKucukHarfClick)
- Arama değiştirme işlemleri (miSearchFindNextClick, miSearchItemClick, miSearchReplaceClick, Ara)
- Yazı rengi değiştirme işlemleri (cbColorBoxChange)
- Yazdırma önizleme işlemleri (miFilePrintPreviewClick)
- Yazdırma işlemleri (miFilePrintClick, YazdirHazirla, YazdirSonlandir,

Yazdir)Tanıma sonucu işlemleri (WMRecognisedText)

- XCekirdek'ten gelen mesajları gösterme (WMShowMessage)

3.5. Konuşma Tanıyan Metin Editörü - KTM

Sistem temel olarak basit özellikler sunan bir metin editörüdür. Sunduğu özellikler WordPad ile karşılaştırılabilir düzeydedir. En önemli fark konuşma tanıma yeteneğine sahip olmasıdır.

3.5.1. Metin editörü

Metin editörü temel Rich Text formatında dosya üreten ve bu dosya formatında bulunan metin niteliklerinin düzenlenmesini sağlayan özellikleri içermektedir.

3.5.2. Konuşma tanıma

Sistemde konuşma tanıma iki ayrı amaç ile kullanılmaktadır.

- Menülerin gezilmesi ve komutların verilebilmesi
- Metin editörü içine mikrofondan okunacak yazı fişlerinin eklenmesi.

Hazırlanan konuşma tanıma teknolojisinin özellikleri aşağıdaki gibidir:

- HTK araç kiti ile akustik model eğitimi yapılmıştır.
- Çok konuşmacılı ses veritabanı kullanılarak konuşmacı bağımsız modeller elde edilmiştir.
- Komut verme esnasında izole sözcük tanıma modunda, fişlerin tanınması esnasında ise sürekli tanıma modunda çalışılmıştır. (Aslında fiş içerikleri kesin olursa, yani belirli sayıda sabit cümle listesi olursa bu da uzun bir akustik modelden oluşan izole sözcük tanıma olarak düşünülmelidir. Ancak fiş tanıma modunun sınırlı sayıda sözcük ile sürekli konuşma

tanıma yapacak şekilde tasarlanması da mümkündür.)

- Komut sözcüklerine ve fiş cümlelerine / sözcüklerine dahili bir arabirim üzerinden veya dışarıdan bir dosya aracılığı ile ekleme - çıkarma yapılması seçeneği olabilmektedir (Tri-phone akustik model kullanımı ile her tür sözcük tanınması mümkündür.).
- N-best listesi çıkarabilen bir konuşma tanıma kodu yazılmıştır (Kullanıcıdan gelen on-line düzeltme arabiriminde yanlış tanınan metin yerine olasılığı en yüksek birkaç metin daha gösterilebilmektedir).
- Ses veritabanı olarak, kaydedilen 1600 adet cümle kullanılmıştır.
- MDT (Morphologic Dynamic Tree) sistemindeki kodlar kullanılmıştır.

3.6. Çalışma Planı

Konuşma tanıma sistemi için yapılması gereken çalışmalar.

- Veritabanı metinlerinin seçimi.
- Ses kayıtlarının toplanması.
- HTK' nın kurulumu ve çalışır hale getirilmesi.
- Akustik modellerin üretimi.
- HTK ile basit model güvenlik testlerinin yapılması.
- Test ses verilerinin elde edilmesi.
- Native kodlarla modellerin test edilmesi.
- Komut tanıma ve fiş tanıma modunda çalışabilecek ve sözcük listesi ve grameri ayarlanabilir konuşma tanıma motorunun hazırlanması.
- Win32 thread içinde çalışabilecek, sürekli mikrofonu dinleyen bir KTM-Konuşma Tanıma modülünün yazılması, konuşma tanıma motorunun modül içine eklenmesi.
- Konuşma tanıma modülünce üretilen komut ve fiş girişlerinin arayüze aktarılmasının sağlanması.
- Ergonomiyi arttıracak AGC (Automatic Gain Control, Otomatik Kazanç Ayarı) vs. ayarların yazılıma eklenmesi (Seçimlik)
- Testler.

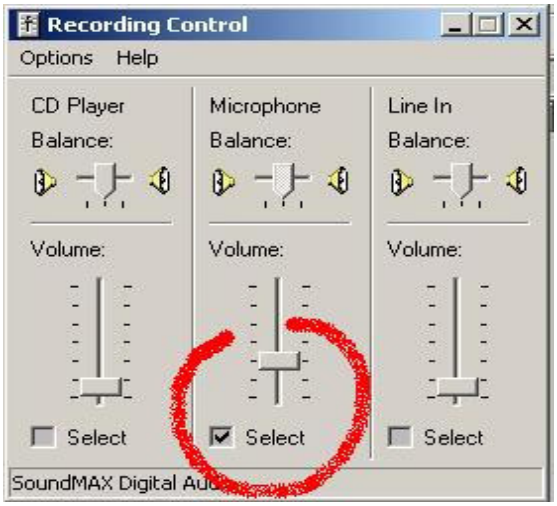
3.7. KTM v(1.1)' nin Konuşma Tanıma Özelliği

KTM'nin en önemli özelliği konuşma tanıma sayesinde metin yazılması ve komut verilmesi işlemlerinde kullanıcıya sesiyle giriş yapma imkanı sağlamasıdır. Konuşma tanıma sistemiyle ilgili bilgiler aşağıdaki başlıklar altında toplanmıştır.

- Mikrofon Ayarları
- Fiş Tanıma
  - Fiş Listesi Düzenleme
- Komut Tanıma

3.7.1. Mikrofon ayarları

Konuşma tanıma özelliklerinin kullanılabilmesi için sistemde bir ses kartı ve buna bağlı bir mikrofon bulunması gerekmektedir. Tercihen mikrofonun düşük gürültü seviyesine sahip olması iyi olacaktır. Sistem ses ayarlarından kayıt sırasında seçili kaynağın mikrofon olması sağlanmalıdır (Şekil 3.3). Ses mikrofon girişi ses yüksekliği de uygun şekilde ayarlanmalıdır. Deneme olarak ses kaydedici ile kayıt yapılarak konuşmanın rahat anlaşılabilir olduğundan emin olunmalıdır.

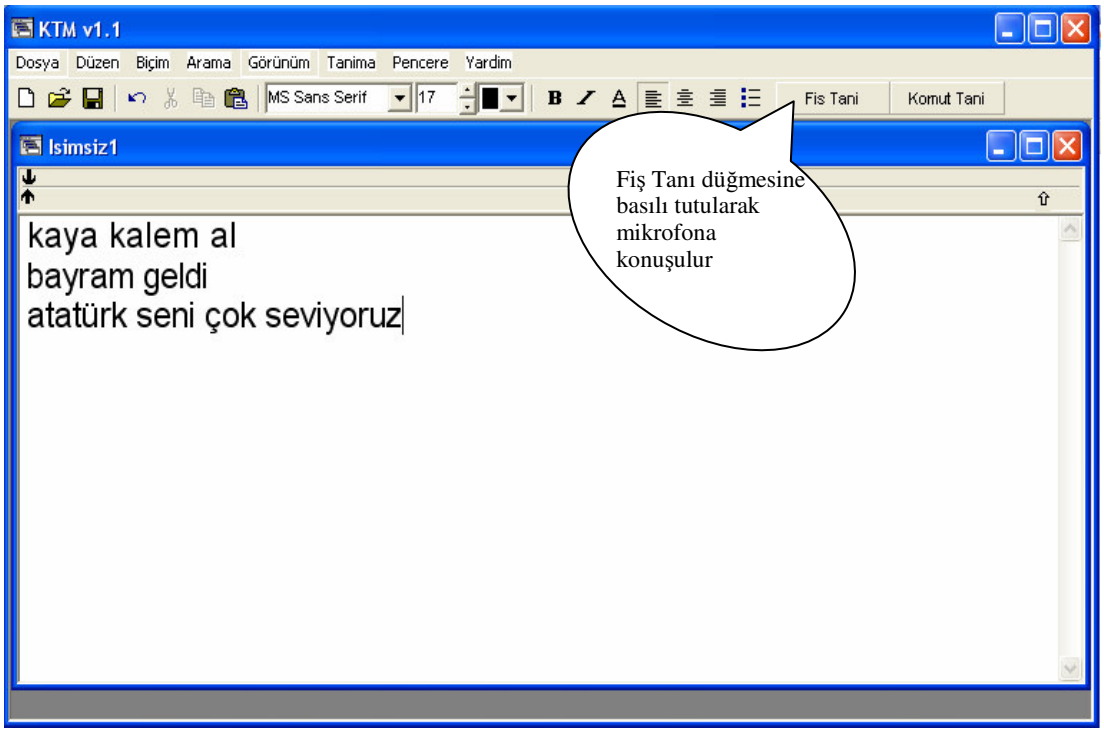


Şekil 3.3. Windows 98 ortamında mikrofon ayarlarının yapılması işlemi

3.7.2. Fiş tanıma

Tanıma çubuğundaki Fiş Tanı düğmesi ile fiş tanıması yapılarak, tanınan fişin seçili pencere içine yazdırılması sağlanmaktadır (Şekil 3.4). Bunun için aşağıdaki adımlar gerçekleştirilmelidir.

1. Fiş Tanı düğmesine tıklamak (farenin sol düğmesini bırakmamak)
2. Mikrofona yazdırmak istenilen fişi söylemek.
3. Fare düğmesini bırakarak tanıma işlemini başlatmak.



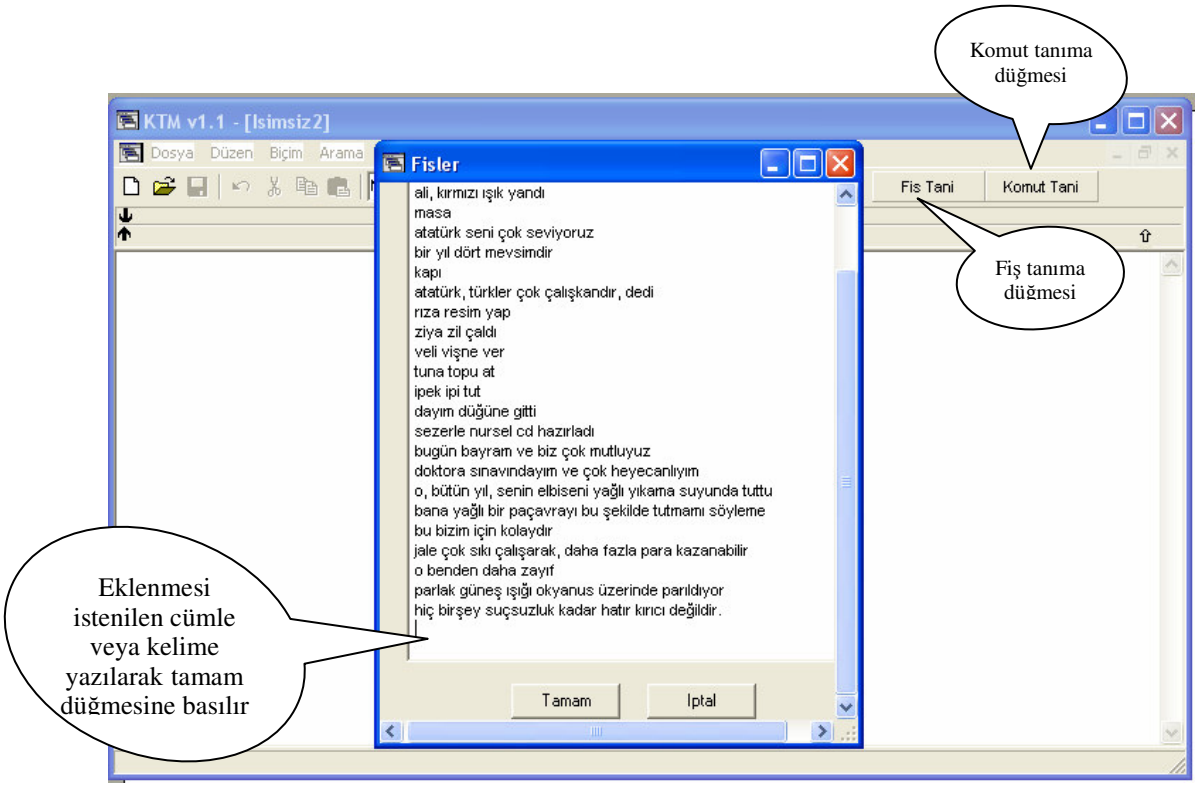
Şekil 3.4. Fiş Tanıma düğmesiyle ekrana yazdırılmak istenilen cümlenin seslendirilme aşaması

Kısa bir süre içinde söylenmiş olan fiş tanınarak metin penceresi içine eklenecektir. Dağarcıktaki fişleri görmek ve düzenlemek için Fiş Listesi penceresi kullanılmaktadır.

Fiş listesi düzenleme

Fiş listesi penceresini açmak için tanıma menüsünden Fiş Listesi seçeneği seçilmelidir.

Pencerede dağarcıktaki fişler liste halinde görüntülenebilmektedir. Liste üzerinde herhangi bir fiş seçilerek istenen değişiklikler yapılabilmektedir. Bu liste üzerinde her satır, bir fişe karşılık gelmektedir (Şekil 3.5).



Şekil 3.5. Fiş Listesine her hangi bir cümle ekleme veya değiştirme işlemi aşamaları

Yapılan değişiklikleri onaylayarak fiş listesini kaydetmek için Tamam, kaydetmeden çıkmak için İptal düğmesi tıklanılır.

3.7.3. Komut tanıma

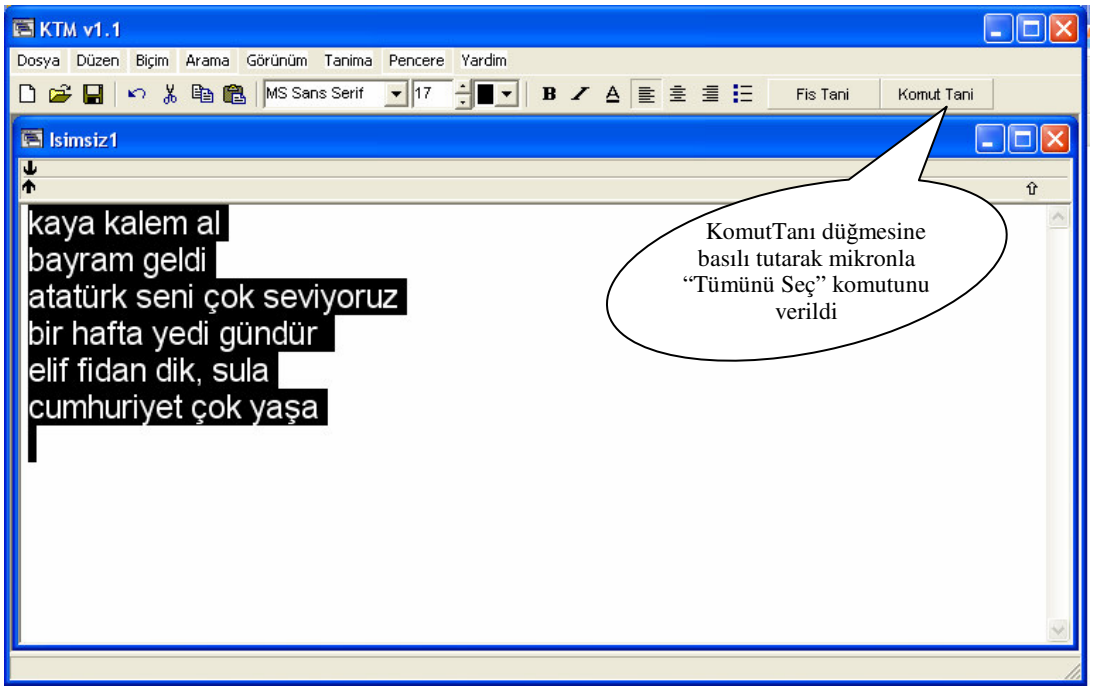
Tanıma araç çubuğu üzerindeki Komut Tanı düğmesine tıklanarak menü komutları uygulanabilmektedir. Bunun için aşağıdaki adımlar

gerçekleştirilmelidir.

1. Komut Tanı düğmesine tıklamak (farenin sol düğmesi bırakılmadan).
2. Mikrofona uygulamak istenilen komutu söylemek.
3. Fare düğmesini bırakarak tanıma işlemini başlatmak.

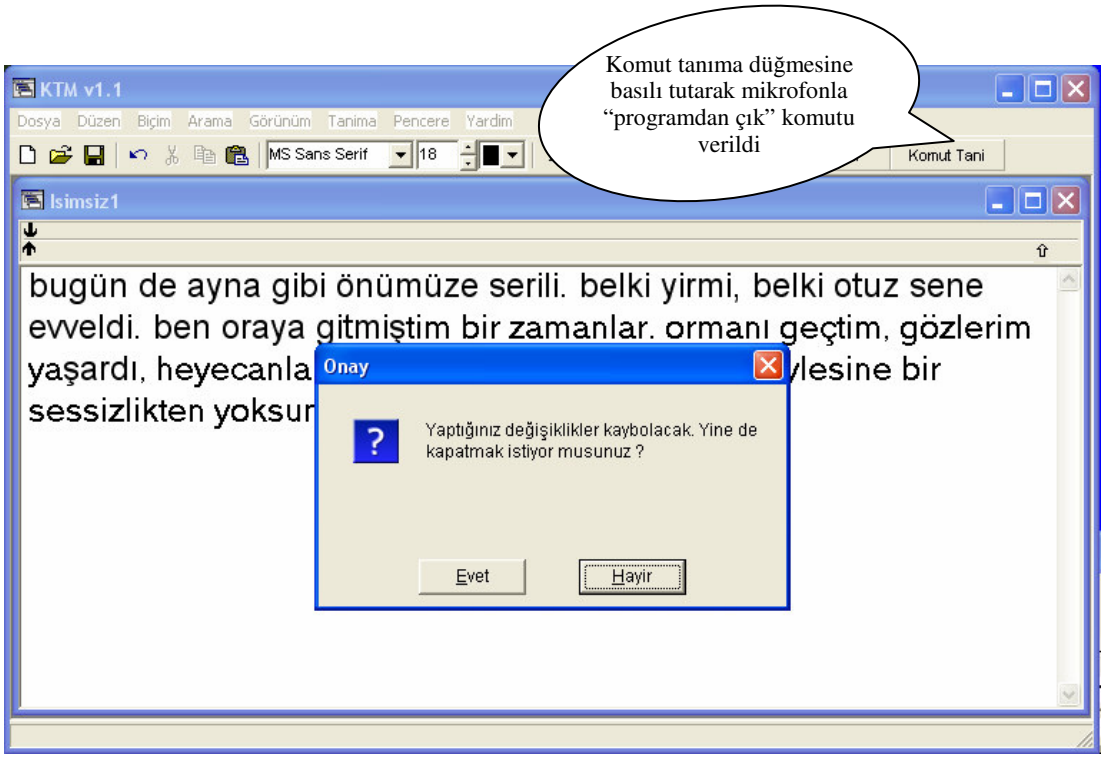
Şekil 3.6 da “tümünü seç” komutunun, Şekil 3.7 de ise “programdan çık” komutunun çalışma düzenleri görülmektedir.

KTM menüsündeki tüm komutlar bu şekilde uygulanabilmektedir. Tanınmış olan komut durum çubuğunda birkaç saniye görüntülenecektir.



Şekil 3.6. Komut Tanı düğmesine mikrofonla “Tümünü seç” komutunun verilmesi

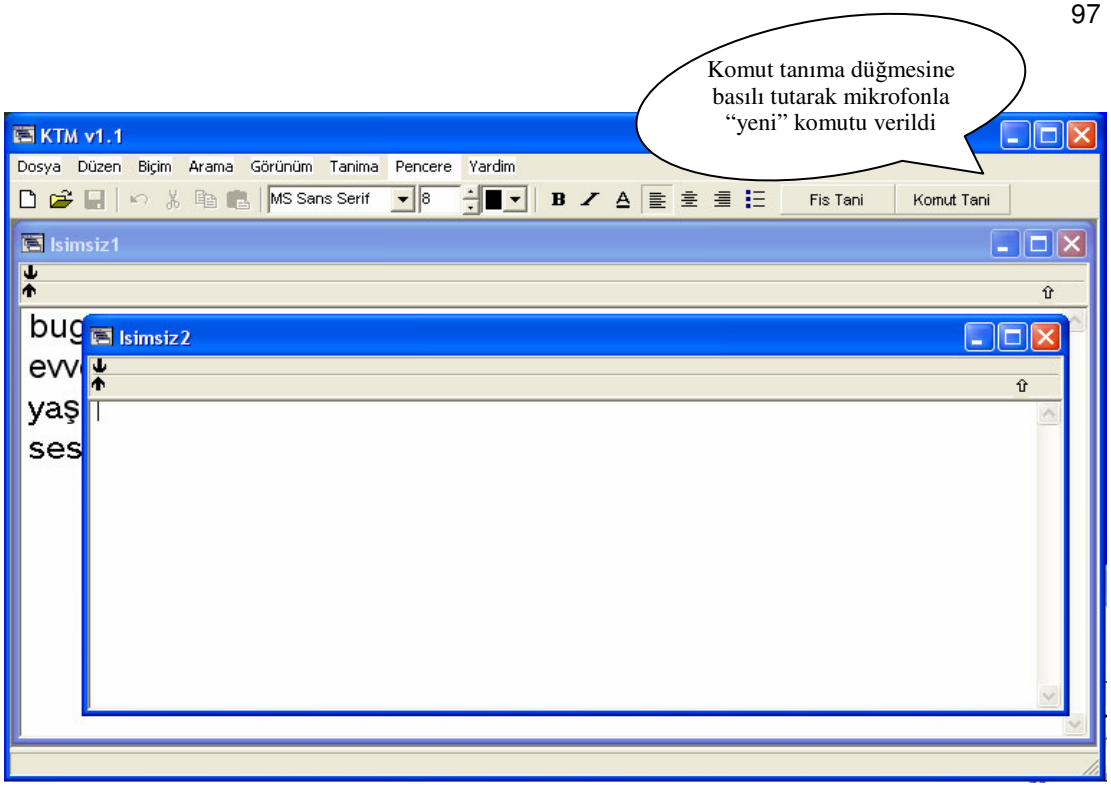
Şekilde de görüldüğü gibi metin editörüne mikrofonla seslendirilen bu cümleler okunulmuş ve Komut tanı düğmesine fareyle basılı tutarak “Tümünü Seç” komutu verilmiş ve de tüm belgenin seçildiği izlenmiştir.



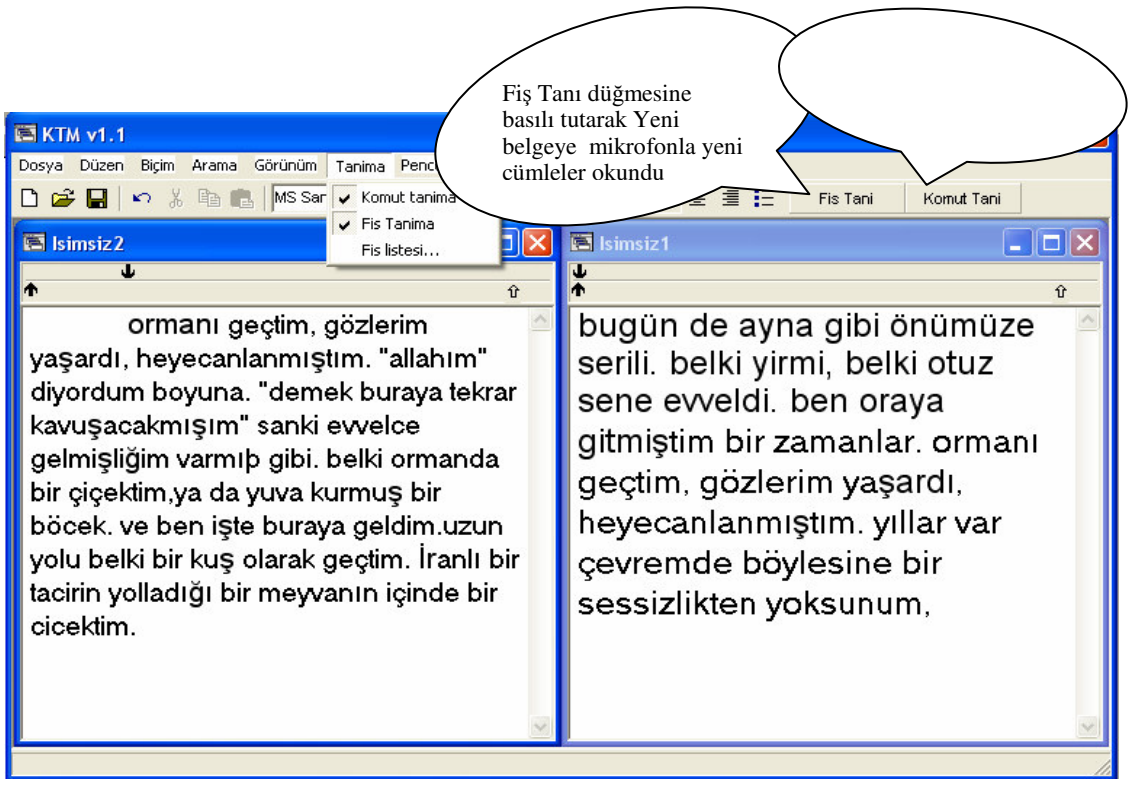
Şekil 3.7. Komut Tanı düğmesine mikrofonla "Programdan Çık" komutunun verilmesi

Aynı şekilde menüler de bulunan tüm komutların bu şekilde uygulanması mümkündür olmaktadır. Fiş Listesinin alabileceği satır sayısı 10.000 kadardır. Yani 10.000 ayrı cümlenin bu sistemde tanınması mümkündür.

Yine bu yazılımla uygulanabilinecek adımların bazıları Şekil 3.8 ve Şekil 3.9 da görülmektedir.



Şekil 3.8. Komut Tanı düğmesine mikrofonla "Yeni" komutunun verilmesi

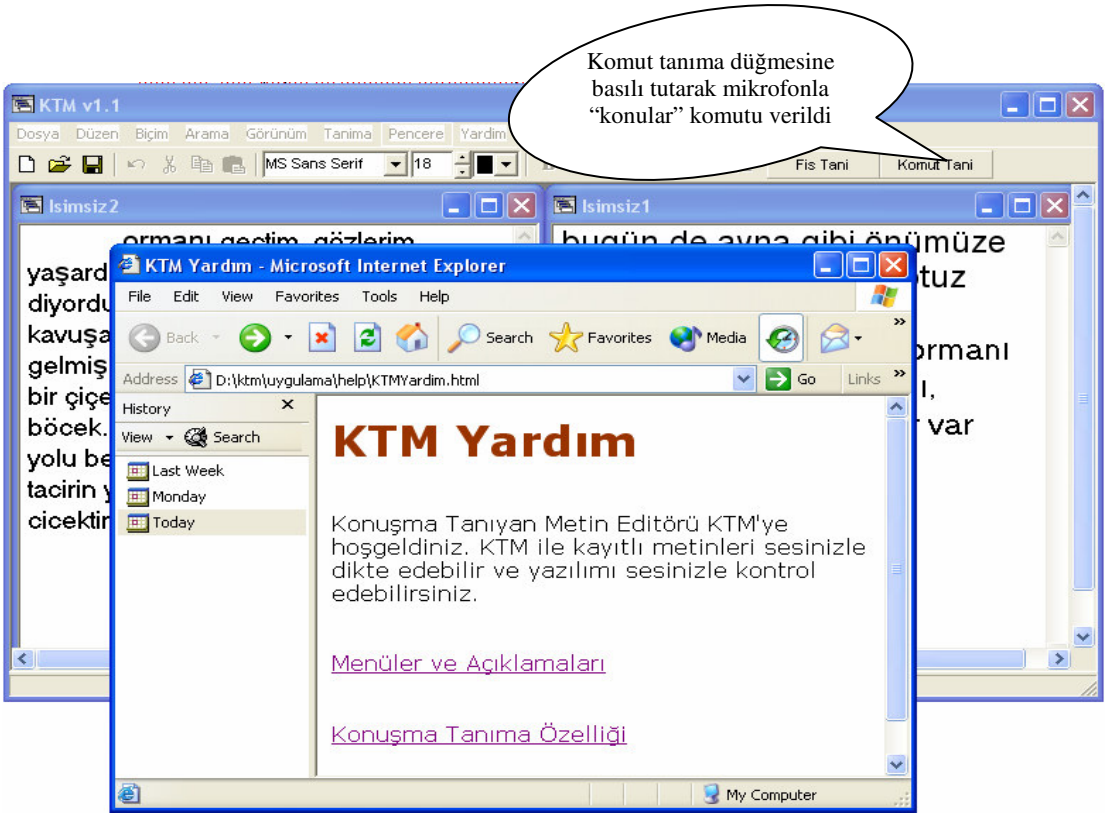


Şekil 3.9. Fiş Tanı düğmesine mikrofonla yeni belgeye cümleler okuma ve Komut Tanı düğmesine mikrofonla "Dikey Sırala" komutunun verilmesi

KTM yazılımında kullanılabilecek sesli komutların tam listesi EK-2 de sunulmuştur. Bunlardan bazıları şu komutlardır;

büyük küçük harf,
tarih saat ekle
metin ara
sonrakini bul
değiştir
araç çubuğu
ses aygıtı
cetvel
komut tanıma

Şekil 3.10 da "yardım" komutu mikrofonla seslendirildiğinde yazılım, KTM hakkında yardım alma konumuna getirilmiş olmaktadır.



Şekil 3.10. KTM metin editörü hakkında bilgi edinmek için "Komut Tanı" düğmesine mikrofonla "yardım" komutunun verilmesi

4. İLKOKUMA YAZMA ÖĞRETİMİ YÖNTEMLERİ ve KTM' YE UYGULANMASI

Günümüzde ilkokuma yazma alanı ile ilgili yapılan bütün sınıflamalarda aşağıdaki yöntemlerden bahsedilmektedir.

4.1. Bireşim (Sentez-Harf-Alfabe) Yöntemi

Bu yöntemde harfler ve harflerin sesleri öğretilerek okuma yazma öğretimine başlanır. Öğretilen harflerin birleştirilmesiyle hece ve kelimeler meydana getirilir (33).

4.2. Çözümleme (analiz-cümle) yöntemi

Bu yöntemde okuma ve yazmaya çocuk seviyesine uygun, anlamlı cümlelerle başlanır. Bu yöntem öykü içerisinden çekilmiş bir cümle ve cümlenin sözcük, hece, ses, hece, sözcük, cümle, öykü devrelerinin aşamalı ve geçişli olarak işlemlerinden oluşmaktadır (33).

4.3. Karma (karışık) yöntem

Bu yöntemde de okuma-yazma öğretimine, cümle yönteminde olduğu gibi cümlelerin verilmesiyle başlanır. Sözcük, hece ve harflerin tanıtılmasına hemen geçilir. Cümle, sözcük, hece ve harf birlikte öğretilir. Cümle yönteminin hızlandırılmış bir şeklidir. Kısa sürede ses ve harflerin öğretimine kaydığı için çocuk psikolojisine ve algılama özelliklerine uygun düşmediği savunulmaktadır. Bireşim yöntemindeki sakınca karma yöntemi için de geçerlidir (33).

4.4. Öykü Yöntemi

Bu yöntemde okuma yazma çalışmalarına anlamlı bütünlüğü olan bir masal ya da öyküden başlanmaktadır. Hazırlanan öykünün cümle cümle

kavratılması ve bu öykü üzerinde yeterince çalışmalar yapılarak cümle yönteminde olduğu gibi sözcük, hece ve harflerin ve bu yol kullanılarak okuma-yazmanın kavratılması amaçlanmaktadır. Bu yöntemle okuma-yazma öğretiminin okuduğunu anlama düzeyini artırdığı savunulmaktadır (33).

4.5. Ses Temelli Cümle Yönteminin Aşamaları (yeni sistem)

Ses Temelli Cümle Yöntemine göre ilk okuma-yazma öğretimi aşağıdaki aşamalar izlenerek gerçekleştirilmektedir (22).

1. İlkokuma-yazmaya hazırlık.
2. İlkokuma-yazmaya başlama ve ilerleme.
   a. Sesi hissettirme ve tanıma
   b. Sesi okuma ve yazma
   c. Sesten heceler, hecelerden kelimeler, kelimelerden cümleler oluşturma
3. Okur yazarlığa ulaşma.

Hazırlanan KTM konuşma tanıyan metin editörü yazılımı yardımıyla ilkokuma yazma işlemleri yapılabilecektir. KTM yazılımında yapılacak bu aşamaların öğrenciler tarafından görülebilmesi için bilgisayarın yanında data show ve tahtaya yansıtılması içinde bir data show perdesi veya bir projeksiyon aleti gerekmektedir. Öğretmen tarafından KTM ile yapılan çalışmalar tahtaya yansıtılarak bütün öğrencilerin bu işlemi takip edebilmesi sağlanmış olacaktır. KTM konuşma tanıyan metin editörü ilkokuma yazma aşamasında kullanılacak bütün yöntemlere uygundur.

Hazırlanan yazılımda fiş listesine tek başına sessiz harfleri eklemek anlamsız olmaktadır. Bunun sebebi ise; sessiz harflerin seslendirilişleri yazılışları ile aynı olmadığı için yalnız kullanılamamaktadırlar. Örneğin, “h”, “ş”, “t”, “z” ve diğer sessiz harflerin tanıtma işleminde mikrofona okuma şekli “he”, ”şe”, ”te”, ”ze” olacağından “h” harfi “he” hecesi ile “ş” harfi “şe” hecesi ile “t” harfi “te” hecesi ile “z” harfi “ze” hecesi ile karışmakta ve de sadece tek başına olan sessiz harflerin tanıma işlemi yapılamamaktadır. Bu örnekleri tüm sessiz

harfler için çoğaltmak mümkündür. Bütün bu sebeplerden dolayı sadece harfler için ve gerekirse heceler için programın fiş tanı işlemi kullanılmadan öğretmen tarafından doğrudan editöre yazılarak yönteme, zamana ve müfredata göre gerekli olan harfler öğrencilere öğretilir.

İlkokuma yazma döneminde hangi ilkokuma yazma öğretimi yöntemi kullanılması gerekiyorsa; ona göre de KTM yazılımı kullanılmalıdır. Öğretmen hecelerle çalışacaksa gerekli heceleri, kelimelerle çalışacaksa gerekli kelimeleri, cümlelerle çalışacaksa gerekli cümleleri fiş listesine eklemelidir. Ama bu işlemleri öğretme zamanına göre yapmalıdır. Çünkü öğrenci de öğretmenin bilgisayarda bulunan bu yazılımın kullanışını takip edecek ve anlamaya çalışacaktır. Yazılımın fiş listesine hece, kelime veya cümle ekleme yazma işlemini, mikrofona seslendirme ise okuma işlemini yerine getirmiş olacaktır.

Çözümleme yönteminin aşamalı uygulanışı her öğrenci için katı bir hiyerarşi göstermemekle birlikte aşağıdaki gibi gösterilebilir.

Hazırlık öykü
Cümle
Sözcük
Hece
Ses
Hece
Sözcük
Cümle
Öykü

4.6. İlkokuma Yazma Öğretiminin Basamakları

Bunlar hazırlık dönemi, cümle çözümleme dönemi, kelime çözümleme dönemi, harf tanıma dönemi ve serbest okuma yazma dönemidir.

4.6.1. Hazırlık dönemi

Öğretmenin ve öğrencinin hazırlığı olarak iki çeşit hazırlıktan söz edilebilir. Öğretmenin hazırlığının en önemli noktası uygulanacak yöntemin ve öğrenme-öğretme etkinliklerinde kullanılacak stratejilerin belirlenmesi ve buna ilişkin hazırlıkların yapılması. Öğrencilerin hazırlığı olarak ise, her şeyden önemlisi öğrencilerin okula uyum sağlama hazırlığı, okuma-yazma için gerekli araç ve gereçleri kullanabilme hazırlığı, okuma-yazma için gerekli araç ve gereçleri kullanabilme hazırlığı, yazmaya hazırlık, okumaya hazırlık sayılabilir (33).

Hazırlık dönemi örnek uygulamaları

Buluş yoluyla öğrenme stratejisiyle gösterip yaptırma (demonstrasyon) tekniği ve soru-cevap tekniği kullanan öğretmenin hedefi yazıyı yazmada elin hareket yönünü kavrayabilme üzerinde durur (33). Buna göre;

Hedef Davranışlar

1. Parmağı sayfa üzerinde soldan sağa doğru hareket ettirme.
2. Deftere yazı yazarken soldan sağa doğru hareket edileceğini söyleme.
3. Defter sayfası üzerinde yazı yazarken hangi yönden hangi yöne doğru hareket edileceğini gösterme.
4. İlkokuma-yazmaya yardımcı olarak kullanılacak olan yazılımdan dolayı basitçe bilgisayar ve klavyesinden bahsetme.

Bir başka ilkokuma - yazma öğretiminde diğer bir hazırlık çalışması şöyledir. Bu kez öğretmen ilkokuma yazmaya hazırlık olan çeşitli düz çizgileri çizebilme ve bilgisayarda gösterebilme hedefi ve alış yolu stratejisiyle ile şu hedef davranışları sağlar:

Hedef Davranışlar

1. Dikey düz çizgi çizme
2. Yatay düz çizgi çizme
3. Eğik düz çizgi çizme
4. Bilgisayar klavyesinden öğretmen yardımıyla dik çizgi (“I” harfi) ve eğik cizgi (“italik “I” harfi)yi yazma veya bunları sadece klavyeden tuşlama

4.6.2. Cümle verme dönemi

Cümle verme dönemi, çocukların bildiği ve kullandığı sözcüklerden oluşan, anlamı kolay kavranılabilecek, çocuğun yakın çevresinden alınmış fiş cümlelerinin, cümle öğretiminin tekniğine uygun olarak verildiği dönemdir (33).

Cümle verme dönemi örnek uygulamaları

Hedef davranışlar

1. “Ali ata bak” cümlesini öykü içinden bulma
2. “Ali ata bak” cümlesini okuma
3. “Ali ata bak” cümlesini havada ve sıra üstünde parmakla yazma
4. “Ali ata bak” cümlesini tebeşirle tahtaya yazmakla beraber, yazılıma mikrofon kullanarak yazdırma(önceden bu cümle öğretmen tarafından programa yazılır), fasulyelerle sıranın üzerine yazma, söyleme
5. Ekrana yansıtılan ve programda yazılı olan “ali ata bak” cümlesini bakarak deftere yazma, söyleme

4.6.3. Cümle çözümleme (kelime tanıma) dönemi

Yeteri kadar cümle üzerinde çalıştıktan ve sınıftaki öğrencilerin çoğunluğu, cümlelerdeki birbirine benzeyen kelimeleri fark etmeye başladıktan sonra cümleler kelimelere bölünmeye başlanır ve kelime tanıma dönemine geçilmiş

olur. Elde edilen kelimelerle yeni cümleler ve cümlelerle metinler oluşturma bu dönemde yapılan temel etkinliktir (33).

Kelime tanıma dönem ile ilgili uygulamalar

"Ömer mısır yer" fiş cümlesini kelimelere ayırma. Buradaki hedef öğrencilerin "ömer mısır yer" cümlesindeki kelimeleri kavrayabilmelerini sağlamadır.
Hedef davranışlar ise:

1. "Ömer mısır yer." fiş cümlesini diğer cümleler arasından seçme (bu cümle ve diğer öğrenilen fiş cümleleri ekrana yansıtılan programda görülmektedir. Konuşma tanıma tekniğiyle mikrofonla öğretmen tarafından seslendirilmiş ve dikte ettirilmiştir.)
2. Fiş cümlesini önce bir bütün olarak sonra da kelimelerini vurgulayarak okuma (Öğretmen bazı öğrencilere de mikrofonla bu fiş cümlesini ve bu cümleye ait kelimeleri okutur).
3. Okunan kelimeler arasına çizgi çizilir. Bu iş programın kendi içinde de yapılabilir.
4. Kelimeleri ayrı ayrı okuma. Öğrencilerin tamamıyla da kelimeler ayrı ayrı okunabilirken yine öğrencilerin zaman zaman bu program içinde mikrofonla kelimeleri okumaları sağlanabilir.
5. Kelimelerin yerlerini değiştirerek okuma (yine program da da bu işlem yapılır)
6. Diğer kelime ve cümlelerle bir araya getirerek yeni cümleler oluşturma (programda da yapma)

Kelime tanıma çalışmasında bilinen kelimelerden yeni kelimeler oluşturabilme hedefi ile öğretmen buluş yoluyla öğrenme stratejisini kullanarak öğrencilerin şu hedef davranışları kazanmalarını sağlamalıdır.

1. Bilinen kelimelerden yeni cümleler oluşturma
2. Oluşturulan cümleleri okuma
3. Oluşturulan cümleleri yazma(deftere yazma, klavyeyle programa yazma)

4. Oluşturulan cümleleri mikrofonla okuma, dikte ettirme

4.6.4. Kelime çözümleme (hece tanıma) dönemi

Öğrenciler kelimeler üzerinde yeterince çalıştıktan sonra kelimeler içindeki heceleri fark etmeye başlarlar. Bu durum hece tanıma çalışmalarına başlamak gerektiğinin bir işaretidir. Bu dönemde elde edilen kelimelerle yeni kelimeler, kelimelerle yeni cümleler, cümlelerle metinler oluşturma çalışması yapılmaktadır. Bu dönemde anlamsız heceleri tek başına öğretmekten ve ezberletmekten kaçınılmalıdır. Öğrencilerden bir kısmının bu devre sonunda okuma yazmayı öğrendiği görülebilir (33).

Kelime çözümleme dönemi ile ilgili uygulamalar

Hedef; ayrılmış hecelerden yeni kelime ve cümleler oluşturabilme.

Hedef davranışlar

1. Fiş zarfından, kesilmiş (programda ise öğretmen tarafından "-" işareti ile ayrılmış) olan hece kelimeleri sıranın üzerine dizme. Programda ise mikrofona hece hece okuma
2. Kesilmiş olan hecelerden anlamlı kelimeler oluşturma (programa bu yeni kelimeleri mikrofonla uygulama)
3. Oluşturulan kelimelerden yeni cümle oluşturma (programa bu yeni cümleleri mikrofonla uygulama)
4. Oluşturduğu cümleyi okuma, bakmadan yazma(öğretmen yardımıyla fiş listesine yeni cümleyi klavye kullanarak yazma ve de mikrofonla bu oluşturduğu cümleyi okuyup ekrana yazdırma)

Diğer taraftan bilinen hecelerden yeni kelimeler oluşturabilme hedefi ile öğretmen şu hedef davranışları sağlamalıdır.

1. Gösterilen heceyi doğru olarak okuma(tahtaya yansıtılan ekrandan)

2. Söylenen heceyi tahtaya yansıtılan ekrandaki fişler arasından bulup gösterme
3. Oluşturulan kelimeleri cümle içinde kullanma. Bunu bir kerede programa uyarlayabilme.

4.6.5. Harf tanıma dönemi

Kelimelerin hecelere bölünmesi üzerinde yeteri kadar çalışıldıktan ve alfabemizdeki içerisine alan yeteri kadar hece öğrenildikten sonra, hecelerdeki harfleri tanıma dönemine geçilir (33). Bu devre sonunda öğrenciler bütün harflerin adlarını ve heceler içerisindeki görevlerini tanımış olurlar. Bu dönemde harfler ayrı ayrı ele alınmamalı, cümle yönteminin gerektirdiği şekilde, heceler içerisinde harf tanıma yolu izlenmelidir (33).

4.6.6. Serbest okuma yazma dönemi

İlkokuma-yazma öğretiminde, çözümleme basamakları tamamlandıktan sonra serbest okuma yazma çalışmalarına geçilir. Bu dönem öğrencinin her türlü materyali rahatlıkla okuyabileceği bir dönem olarak algılanmamalıdır (33). Öğrenciler bu dönemde bildikleri kelimelere dayalı basit metinler üzerinde çalıştırılmalı, metinlerin zorluk derecesi bilinçli olarak artırılmalıdır. Bu döneme geçiş okuma-yazma çalışmalarının bittiği şeklinde yorumlanmamalı, bu dönem de en verimli şekilde değerlendirilmelidir. Özellikle okuduğunu anlama üzerinde durulmalıdır (33).

<u>Serbest okuma yazma dönemine yönelik bir uygulama</u>

Basit bir tekerleme ya da metin oluşturma hedefiyle öğretmen öğrencilerde şu hedef davranışları gözlemlemelidir.

1. Öğrenilmiş olan kelime ve heceler yardımıyla yeni cümleler oluşturma. (öğretmen bir öğrenciye yardımcı olmak koşulu ile bu yeni cümleleri programımızın fiş listesi kısmına ekleterek yani klavye kullanarak

bilgisayara yazma imkanı sağlamalı ve her farklı çalışmada farklı öğrenciyle bu tür çalışmaları yaparak onların klavye kullanma alışkanlıklarını da sağlamalıdır. Böylelikle öğrenci klavyedeki harflerin yazılışlarını üzerine basarak gerçekleştireceğinden pekiştirme fırsatı da yakalamış olacaktır.)

2. Oluşturulan yeni cümlelerden tekerleme ya da metin yapma
3. Tekerleme ya da metne başlık bulma
4. Tekerleme ya da metni okuma, yazma. ve öğretmen yardımıyla bu tekerlemeleri programı kullanarak klavye vasıtasıyla yazma ve de mikrofon kullanarak ve bunları okuyarak ekrana yazdırma. Böylelikle konuştuğunun anında yazıya dökülmesi ve nasıl yazıldığını anında ekranda görmesi öğrencinin hızlı kavrama yetisini artıracaktır.

Örneğin; öğretmen öğrencilerle beraber bir çalışma yaparak öğrenilen kelime ve hecelerden konu başlıkları belirleyerek fikirler alır. Ve bu konu başlıkları programımızın fiş listesine eklenerek mikrofonla okunup ekrana yazılması sağlanır. Bu arada birkaç değişik tekerleme oluşturmak ve öğrencileri aktif bir şekilde derse katılımını sağlamak adına 3 (isteğe göre daha da arttırılabilir) gruba bölebiliriz.

Ortaya çıkan konu başlıkları (33);

- Ağaç	- Elma	Seçilen Konular
- Soba	- Yeşil	birinci Grup : - elma
- Okul	- Güzel	ikinci grup : - top
- Top	- Oyun	üçüncü grup : - kalem
- Bahçe	- Kuş	
- Kalem	- Çanta	

Öğretmen öğrencilere her grubun kendi aralarında çalışarak seçilen konularla ilgili metin oluşturmalarını istemelidir.

Çizelge 4.1. Üç farklı öğrenci grubunun oluşturdukları metin

1. grup	2. grup	3. grup
Elma	Top	Kalem
Babam elma aldı. Elma çok tatlı. Çok güzel elma. Sakın kaçırma.	Bak bu top. Hop, hop, hop. Sarı top. Sarı topu tut. Topu bana at.	Bu kalem benim. Güzel bir kalem Ne güzel kalem. Ömer kalem al

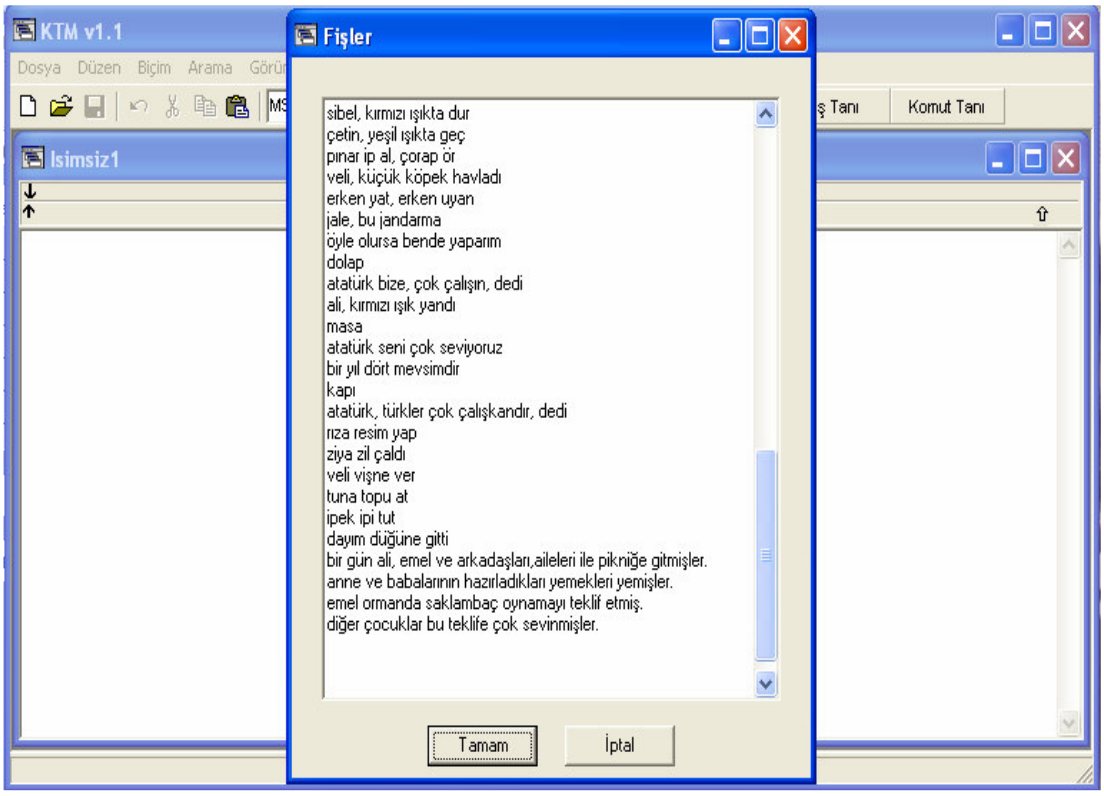
Sırasıyla her grup temsilcisi öğretmenin yardımıyla bu oluşturdukları ve deftere yazdıkları cümleleri öncelikle programın fiş listesine ekler. Sonrada mikrofonla seslendirip okuyarak ekrana yazdırır. Bu çalışmalar panoya asılarak öğrenciler takdir edilerek ödüllendirilir (Ayrıca bu işlem komut tanı düğmesine tıklanarak "yazdır" komutuyla yazıcıya gönderilir. Çıktısı alınan bu yazılar da panoya asılabilir.).

Öğretmen için KTM uygulama örneği yapacak olursak; Çözümleme yöntemine örnek olması açısından öğretmen öncelikle öğrencilere hangi cümleyi öğretmek istiyorsa bu cümleyle ilgili bir öykü anlatır. Öğretilecek cümle "Ali ata bak" olsun. Öğretmen ormanda piknik yapan çocuklar ve etrafta bulunan hayvanlara (at) ilişkin bir resmi tepegözle yansıtır. Öğrencilere;"şimdi size bu resimle ilgili bir öykü anlatacağım" der (33). Öyküyü anlatmaya başlar. Buna uygun öykümüz de şu şekilde olsun. "bir gün Ali, Emel ve arkadaşları, aileleri ile birlikte pikniğe gitmişler. Anne ve babalarının hazırladığı yemekleri yedikten sonra Emel ormanda saklambaç oynamayı teklif etmiş. Ali bu oyunu çok sevdiği için hemen kabul etmiş. Diğer çocuklarda ellerini çırparak bu teklife çok sevindiklerini belirtmişler.

Fikir Emel'den çıktığı için Emel ebe olmuş. Diğer çocuklar da ormanda saklanmışlar. Emel arkadaşlarını ararken, kocaman bir at görmüş. Ali..! ata bak! diye Ali'yi çağırmış"

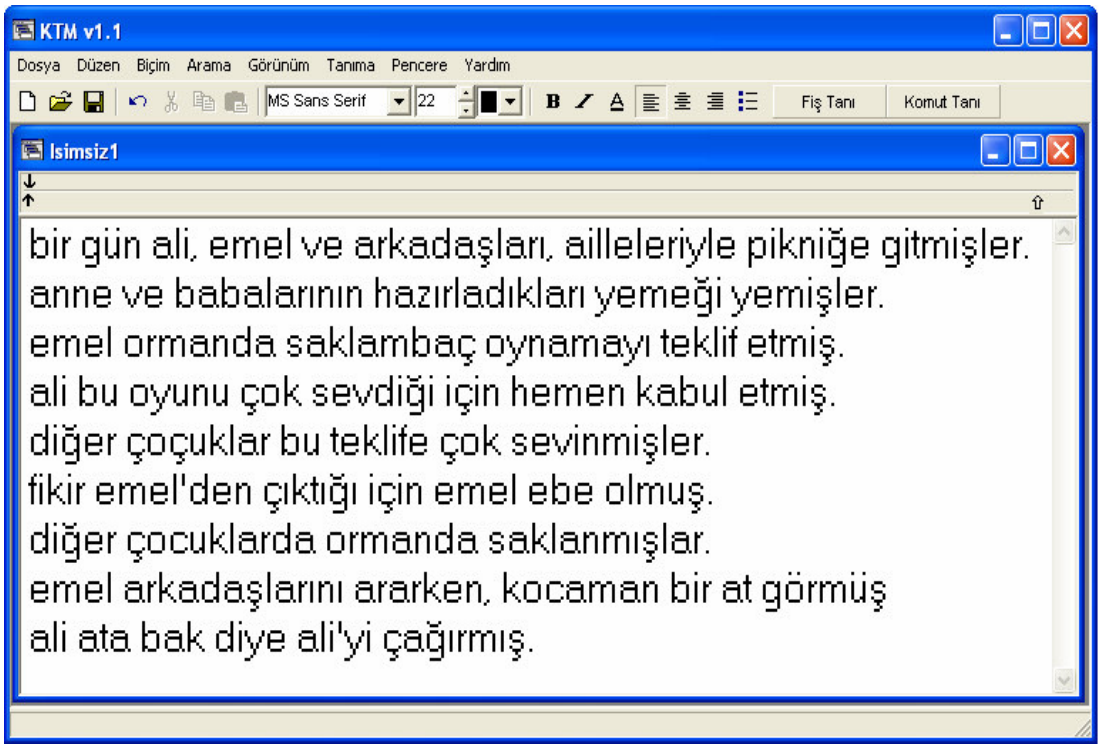
Çocuklar şimdi biraz düşünelim. Acaba öykümüzün devamında neler olmuş. diyerek öykü öğrencilere tamamlattırılır. Öykü içinde geçen “ali ata bak” cümlesi öğrencilere buldurulur (örneğin; Çocuklar, Emel atı görünce ne yaptı? Gibi sorular yöneltilebilir.). Daha sonra öğretmen; çocuklar bugün “ali ata bak “cümlesinin nasıl yazılıp okunduğunu öğrenelim mi? Diyerek öğrencilere sorulur (33).

Öğretmen bu aşamaları KTM yazılımında uygular. Bunun için bir defaya mahsus olmak üzere öğretmen bu öyküyü cümle cümle programımızın fiş listesine eklemelidir (Şekil 4.1). Sonra öğretmen dilerse öğrencilere soracağı soruları da bu yazılımı kullanarak yapabilir. Bu şekilde hem soracağı sorunun yazılışlarının nasıl olduğunu da görmüş olurlar.



Şekil 4.1. Hikayedeki cümleleri fiş listesine ekleme

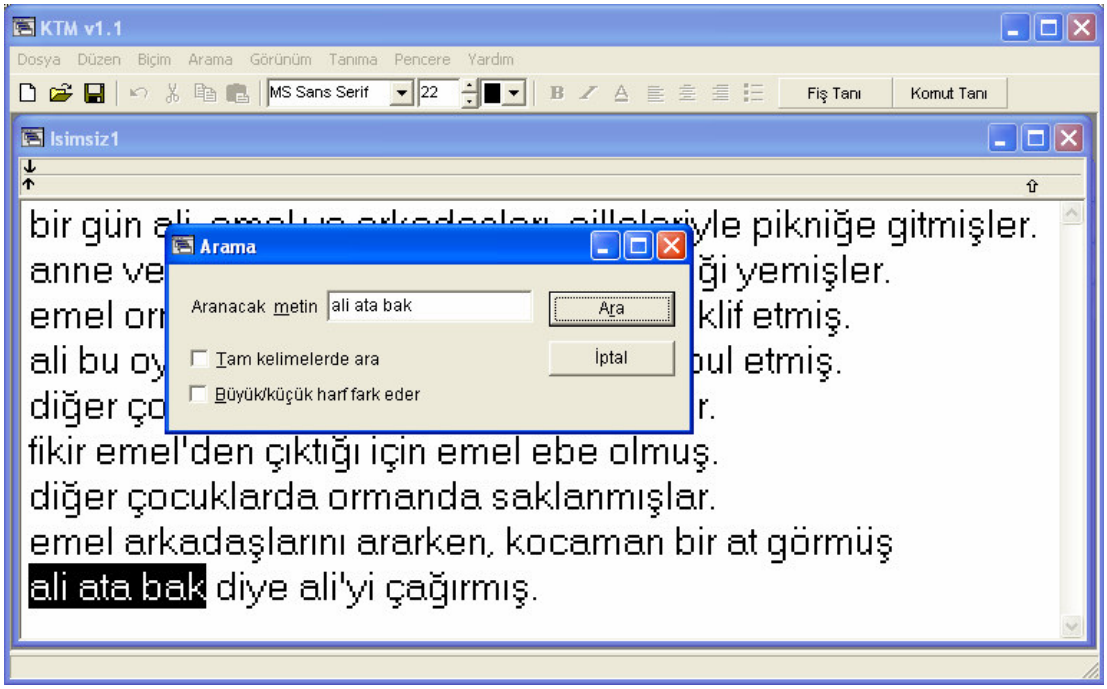
Sonra öğretmen, öğrencilere öyküyü cümle cümle fiş tanı düğmesiyle mikrofona okuyarak yazlımın bu cümleleri tanımasını sağlar (Şekil 4.2).



Şekil 4.2. Hikayedeki cümleleri mikrofonla seslendirerek ekrana yazdırma

Daha sonra da öğretmen bu öykü içinde “ali ata bak” cümlesini komut tanı düğmesiyle mikrofon kullanarak “metin ara” komutunu verir ve hikaye içinde buldurur (Şekil 4.3).

Bu şekilde anlatılan hikaye içinde öğretilmek istenen cümle veya kelime bulunmaya çalışılır.



Şekil 4.3. “Ali ata bak” cümlesinin hikaye içinden “metin ara” komutuyla bulunması

Böylelikle bir öykü içerisinden bir cümle bulunmuş olur. Öğrenciler bu şekilde buldukları cümlenin fiziksel olarak nasıl yazılmış olduğunu ekranda da görmüş olurlar. Öğrencilerin dikkatleri bu cümle üzerinde yoğunlaşır. Öğretmen öğrencilere aynı yöntemi uygulayarak farklı farklı cümleler öğretebilir. Öğretmen öğrencilerin dikkatini çekerek hedeften (öğretilmesi düşünülen cümle) haberdar etmiş olur. Bu durum ise öğrenme-öğretme etkinlikleri içerisinde bulunan bir çalışmadır.

Bu yazılım içerisinde kullanılan dil doğru telaffuz edilmelidir. Bundan dolayı öğretmen de öğrenci (seslendirme kısmında kullandığında) de bu yazılımı kullandıklarından doğru telaffuz etmek zorunda olacaklarından Türkçeyi doğru kullanma ve söyleme durumunda kalacaklardır. Bu da o yaştaki çocuklara öğretilebilecek en iyi hedef davranışlardan biri olacaktır. Türkçeyi doğru okuma ve okuduğunun anında yazıya çevrilmesi ile doğru yazma alışkanlıkları elde etmiş olacaklardır.

Öğretmenin bundan sonra yapacağı ise bu cümleyi parçalayarak öğrencilere sözcük olarak öğretir. Böylelikle öğrencilerin kelimeleri tanıması sağlanmış olur. Yeteri kadar cümle üzerinde çalıştıktan ve sınıftaki öğrencilerin çoğunluğu, cümlelerdeki birbirine benzeyen kelimeleri fark etmeye başladıktan sonra cümleler kelimelere bölünmeye başlanır ve kelime tanıma dönemine geçilmiş olur. Elde edilen kelimelerle yeni cümleler ve cümlelerle metinler oluşturma bu dönemde yapılan temel etkinliktir. Aynı zamanda yeni cümleler verilmeye de devam edilecektir (33). Öğretmen “ali ata bak” cümlesini şu şekilde kelimelere ayırır (programda ise Ali-ata-bak şeklinde belirtir).

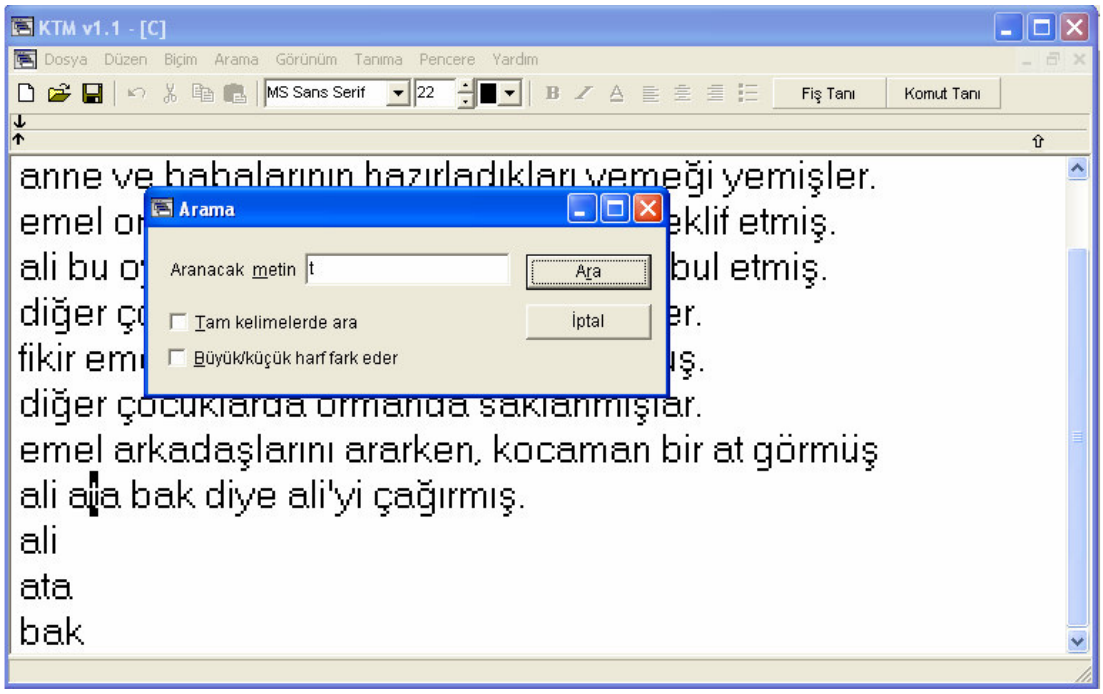
ali
ata
bak

sonra bu sözcükleri öğretmen yazılımı kullanarak yine mikrofon aracılığıyla ekrana yazdırır. Arkasından öğretmen hecelere geçer (programda a-li, a-ta, bak şeklinde belirtir).

a
li
ta
bak

hecelerden sonra öğretmen harflere geçer. Harfleri tek başına programa tanıtmak mümkün değil demiştik. Ancak öğretmen metin ara komutuyla cümle içerisinde geçen harfleri buldurarak öğrencilere gösterir ve öğretmen bu harfin yazılış ve okunuşunu öğretir.

Örneğin; t harfi için; metin ara komutunu verdikten sonra aranacak metin kısmına t yazar ve o an açık olan belge içerisinde t harfini bulur (Şekil 4.4).



Şekil 4.4. “t” harfinin mikrofona “metin ara” komutu seslendirilerek bulunması

Kalıcılığı ve transferi artırmak amacıyla daha önce okunup yazılması öğrenilmiş olan “ali ata bak” cümlesinden yeni bir metin oluşturulur. Bu metin yazılıma okunur ve ekrana yazdırılır. Bu işlemleri öğrenci seviyesine uygun olan cümleleri kullanarak yapmak mümkündür. Buradaki hedef öğrenilmiş olan cümlelerden metin oluşturabilmektir. Böylelikle hedef davranış olarak öğrencilerin bu cümleleri tekrar oluşturma, ayrıca okunan cümlelerden uygun olanlarını alt alta getirerek metin oluşturma, oluşturulan bu yeni metne bir başlık bulma ve metin okuma ve yazma olacaktır.

Böylelikle işbirliğine dayalı bir öğrenme stratejisi kullanılarak öğrencilere yaparak yaşayarak öğrenme sağlanmış olur. Bunun için öğrencilerin birlikte çalışabilecek oturma düzenleri sağlanır (kümelerde her düzeyde öğrencinin bulunmasına dikkat edilir.). Öğrenciler, fiş dosyalarında bulunan küçük fişleri sıranın üzerine dizerler. Öğretmen yapılacak çalışma ile ilgili açıklamalarda

bulunur. Fiş cümlelerinden uygun olanlar alt alta getirilerek bir metin yapılacağı söylenir (33). Fişlerin tamamı okunduktan sonra yan yana oturan öğrencilerden önce biri sonra da diğeri bu fiş cümlelerinden uygun olanlar ile bir metin oluşturur. Biri metindeki kelimelerden metne başlık bulur. Öğretmen ve gruptaki diğer arkadaşları bu öğrenciye "çok iyi" gibi ifadeler söyleyerek yardımcı olarak onu pekiştirici ve güdüleyici sözlerle destekler. Öğrenci alkışlanarak kutlanır (33). Örneklerle bu açıklanacak olursa. Önce öğrencilere öğretilmiş cümleleri sıralayalım.

Ali bak.
Emel eve gel.
Ali ata bak.
Emel bal al.
Bayram geldi.
Resim yap.

Birinci öğrenci (Ali)	İkinci öğrenci (Emel)
Ali bak.	Emel eve gel.
Ali ata bak.	Emel bal al.
Bayram geldi.	Bayram Geldi.
Resim yap.	Resim yap.

Yapılan metinler öğretmen tarafından kontrol edilir ve "çalışmayı doğru tamamladınız, metinlerinizi defterlerinize yazabilirsiniz" şeklinde dönüt verilir. Metinler öğrenciler tarafından defterlere yazılarak çalışma tamamlanır (33).

Öğrenciler daha önce cümlelerden kesilmiş olan kelimelerle yeni kelimeleri bir araya getirerek yeni cümleler oluştururlar (33).

Örnek

Önceden kesilen cümleler:

Ali - bak - .
Ali - ata - bak - .
Emel - eve - gel - .
Emel - bal - al - .
Yeni oluşturulan cümle
Ömer - mısır - yer - .

Öğrencilerce oluşturulabilecek cümleler

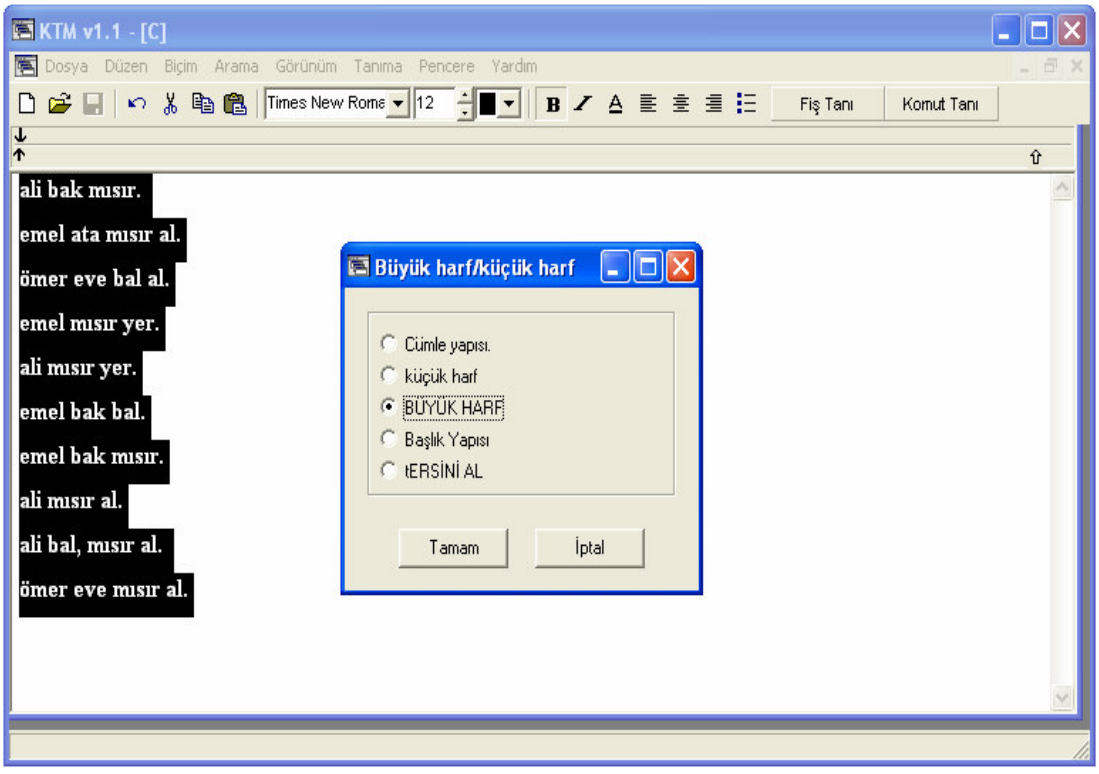
Ali bak mısır.
Emel ata mısır al.
Ömer eve bal al.
Emel mısır yer.
Ali mısır yer.
Emel bak bal.
Emel bak mısır.
Ali mısır al.
Ali bal, mısır al.
Ömer eve mısır al.

Bu oluşturulan cümleler yazılıma eklenir.

Yine yazılımdan bu cümlelerin tanınması beklenir. Her bir öğrenciye bu cümleleri mikrofona seslendirerek tekrarlamaları sağlanır. Böylelikle öğrenci düzgün telaffuzla bu cümleyi mikrofona okumanın sevinci ile daha dikkatli ders takibi içinde bulunacak ve ayrıca bağımsız alıştırma yapabilmenin heyecanını yaşayacaktır.

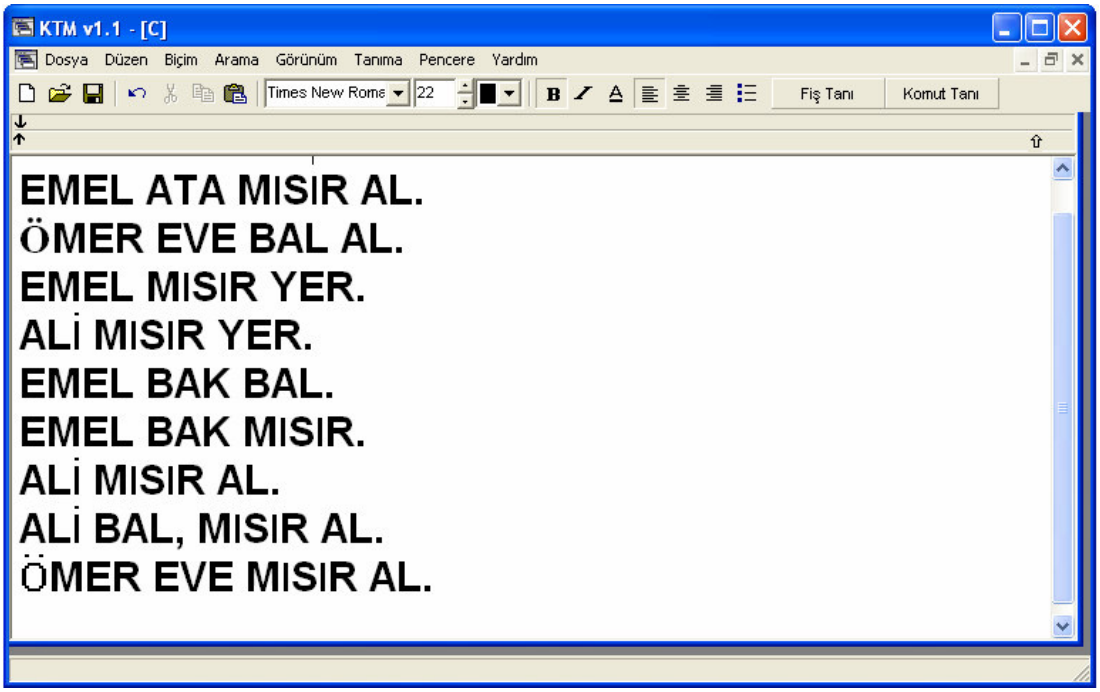
Öğretmen öğrencilerle bir diğer taraftan bilinen kelimelerden yeni cümleler oluşturmaya çalışır. Büyük harflerin tanıtımı söz konusu olunca mikrofonla tanınması istenen cümlelerin büyük harfe dönüştürülmesi de mümkün bu yazılımla. Bunun için de yapılması gereken komut tanı düğmesini kullanarak "büyük küçük harf" komutu mikrofonla verilir.

Karşımıza gelen pencereden Büyük harf sekmesi seçilir ve tamam düğmesine basılır (Şekil 4.5).



Şekil 4.5. Öğretilen cümle ve cümleleri küçük harften büyük harfe dönüştürmek için "komut tanı" düğmesi kullanılarak mikrofonla "büyük küçük harf" komutu uygulaması

Elde edilen sonuç Şekil 4.6 da görülmektedir.

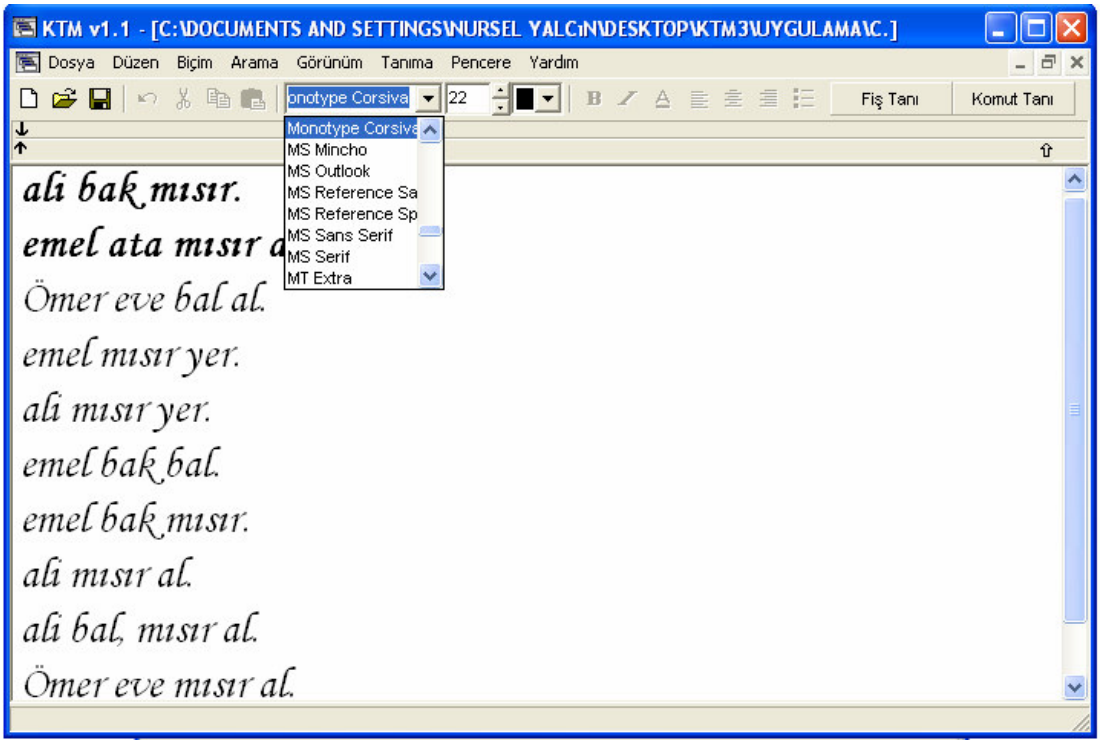


Şekil 4.6. Küçük harfle yazılmış cümlelerin sesli komutla büyük harfe dönüştürme

Ses temelli çalışan yeni eğitim sisteminde bitişik eğik el yazı uygulaması söz konusudur.

Bunun içinde öğretmen yazı fontunu Monotype Corsiva olarak ayarlamalıdır. Metni farklı bir el yazısı yazı tipine çevirmek de mümkün olmaktadır.

Buna göre KTM deki görünüm Şekil 4.7 deki gibidir.



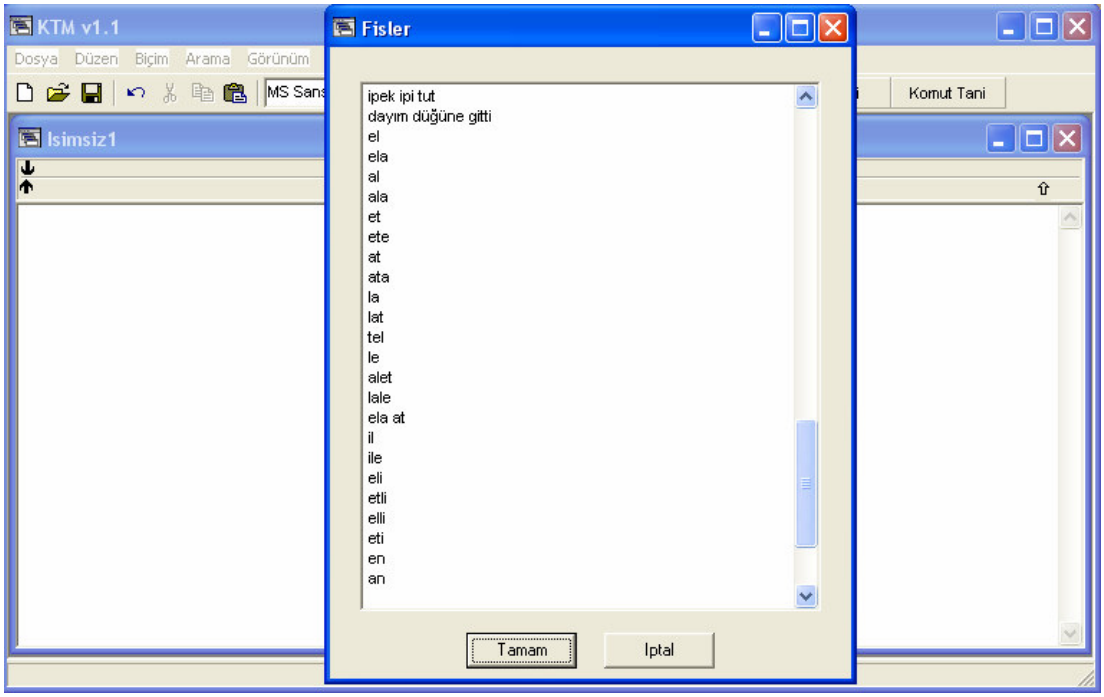
Şekil 4.7. Eğik yazı biçiminin programdaki uygulaması

Öğretmen fiş listesine yeni eğitim sistemine uygun birinci grup sesler olan e, l, a, t harf dizisine göre çıkarılabilecek hece ve sözcükleri ve bu bunlardan oluşacak cümleleri ekler. Bunlar;

el
ele
al
ala
et
at
ete
ata
lat
tel

alet
la
ta
ela
lale
ela al
ela lale al
ela lale elele
ela atla
ela et al
Talat alet al
Lale ata el ele
Lale tel al
El ele
Ela at

Bu ifadeleri KTM yazılımı ile fiş listesine ekleme Şekil 4.8 de gösterilmektedir.



Şekil 4.8. Hecelerin programın fiş listesine eklenmesi

5. SONUÇ VE ÖNERİLER

Bu tezde konuşma tanıma sistemlerinde farklı bir çalışma alanı olarak eğitim ele alınmıştır. KTM yazılımı, konuşmacıdan bağımsız çalışan geniş dağarcıklı Türkçe sürekli konuşma tanıma yapabilen, sesli komutlarla programı kontrol edebilen bir metin editörü olarak, ilköğretim birinci sınıf öğrencileri için ilkokuma-yazma sürecinde kullanmak için geliştirilmiştir.

Bu çalışmada tanınması istenen cümleler ilköğretim birinci sınıfta öğrencilere öğretilen fişleri kapsamaktadır. Fakat KTM Konuşma Tanıyan Metin Editörü içinde oluşturulan program sözlüğüne (fiş listesi) eklenen herhangi bir sesli ifade de yazılım tarafından tanınmaktadır (geniş dağarcıklı tanıma). Bu işlem konuşmacıdan bağımsız olarak gerçekleştirilmiştir. Model eğitiminde kullanılan ses verileri 16 kişiden alınan toplam 1600 adet cümleden oluşmaktadır. Hazırlanan yazılımın temel iki bileşenlerinden biri olan konuşma tanıma sistemi; mikrofondan söylenen komut veya metin cümlesini (fiş) yazılı metne çeviren modüldür.

Bu çalışmada kullanılan konuşma tanıma sistemi şu özelliklere sahiptir:

- Akustik model olarak üçlüses (tri-phone) modeller kullanılmıştır. Geniş ve esnek dağarcık için bu seçim zaruri olmuştur. Kelime modelleri dağarcıktaki her eleman için ayrı bir eğitim gerektirmiştir.
- Akustik modeller konuşmacı bağımsız olarak üretilmektedir. Böylece Türkçe konuşabilen tüm kullanıcılar (özel ağız ve lehçeler ayrı tutulmak koşulu ile) KTM yazılımını kontrol edebileceklerdir. Bu özelliğin sağlanması için akustik model üretiminde kullanılan ses veri tabanı uzun ve çok sayıda konuşmacıyı içerecek genişlikte olmak durumunda kalmıştır. Tanıma başarımının konuşmacı bağımlı sisteme göre daha düşük olması beklenmelidir (örüntü tanıma sistemi olarak düşünüldüğünde ayırım yapılacak sınıfların karmaşıklık derecesi artırılmış olmaktadır.)

- Kullanılan SMM' lerin eğitimi uzun süredir yaygın kullanıma sahip olan HTK (Hidden Markov Toolkit) araç kiti ile yapılmış, ancak bu kitin kullanımı için gereken tüm yardımcı program ve veriler tarafımdan hazırlanmıştır.
- Sistem tanıma işlemini ağaç yapısına sahip bir SMM ağı üzerinden gerçekleştirmektedir. Bu da doğrusal tanıma yapan kod çözücülere (decoder, tanıma işlemini gerçekleştiren modül) göre (örneğin HTK içindeki kod çözücü) büyük bir hız ve bellek üstünlüğü sağlamaktadır.
- Komut tanıma ve fiş tanıma işlemleri iki ayrı model ağı üzerinden gerçekleştirilmektedir.
- Saklı markov modellerine ait bazı parametreler şunlardır:
  - Öznitelik: Cepstrum
  - Durum sayısı: 5
  - Sürekli Gaussian karışım sayısı: 5

Hazırlanan yazılımın temel iki bileşenlerinden bir diğeri de metin editörüdür. Metin editörü konuşma tanıma modülünün işlev gösterebileceği en uygun örnek ortamlardan biridir. Konuşma tanıma modülünden gelen metin bilgisi hem komut olarak (örneğin menü komutu olarak) hem de yazı içeriği olarak kullanılabilmektedir.

Bu sistemde esas olan nokta konuşma tanıma modülü ile yazılım arayüzü arasındaki bağlantıların kurulmasıdır. Günümüz yazılım teknolojilerinde grafik arayüze sahip yazılımlar olay yönetimli (event-driven) olarak çalışmaktadırlar. Klavyede bir tuşa basılması, fare tıklaması yapılması gibi "olay" lar yazılım tarafından incelenerek gerekli tepkiler oluşturulur. Konuşma tanıma sisteminin metin editörüne bağlanması için şu işlemler gereklidir:

- Konuşma tanıma için ses kaydının yapılması
- Ses kaydı yapıldıktan sonra sesin tanıma modülüne gönderilmesi
- Modül tarafından tanınan sese ait metnin; fiş ise metin klavyeden yazılmış gibi o anda seçili olan pencereye gönderilmesi komut ise; ilgili kullanıcı komutunun gerçekleştirilmesi.

İlk okuma-yazma çalışmalarında dinleme, konuşma, okuma, yazma ve görsel okuma-görsel sunu birlikte ele alınmalıdır. Bütün çalışmalar, Türkçe öğretim programının 1. sınıf düzeyindeki kazanımların gerçekleştirilmesine yönelik düzenlemelerdir.

Bu çalışmada öğrenci yazı ile konuşma arasındaki benzerlikleri görecek, yazının harflerin birleşmesiyle yapıldığını anlayacaktır. Bu yöntemle hazırlanan yazılımın ilkokuma yazma öğrenme sürecinde öğrencilerin bireysel, zihinsel ve sosyal gelişimlerine katkı sağlayacağı düşünülmektedir. Öğrenme ortamında aktif öğrenci, bireysel ve toplumsal gelişimini istenilen yönde daha çabuk geliştirebilir. Gelişmiş bireyler olabilmenin ön koşullarından biri öğrenme ortamında aktif olabilmedir. İlkokumaya bu önemin verilmesi bu çalışmalarda daha dikkatli ve bilinçli olmayı gerektirir.

Konuşma tanıma teknolojisi yardımıyla ilkokuma yazma dönemindeki öğrenci, bilgisayar çağı dediğimiz şu zamanda bilgisayarla tanışmış olarak modern bir eğitim sistemine dahil edilmiş olacaktır. Harfleri öğrendikçe klavyedeki her bir tuşta yazılı olan harfleri özdeşleştirecek ve daha kolay bir şekilde harfleri ve sesleri tanıyacaktır. Günümüzde bazı kelimeleri düzgün telaffuz edemediğimiz malum. Buna sebep içinde bulunulan sosyal ve coğrafi yaşam. Öğrenci bu KTM programında mikrofona seslendirilen sesli ifadeleri düzgün söylemek zorunda olduğundan Türkçe' yi doğru telaffuz etmeyi de öğrenecek ve bunu bir tutum olarak benimseyip yaşamı boyunca doğru telaffuza dikkat edecektir. Aynı durum öğretmen içinde söz konusudur. Çünkü KTM öncelikli ve daha yoğun olarak öğretmen tarafından kullanılacaktır. İlkokuma yazma döneminde öğretmen önemli bir etkendir. Bu nedenle öğretmen bu çalışmalara iyi hazırlanmış olmalıdır. Öğretmen yardımıyla öğrenci mikrofona okuduğu cümleyi veya kelimeyi aynı anda ekranda da göreceğinden seslendirdiği ifadenin yazılışını da erken kavramış olacaktır. Bilgisayarla, mikrofonla, data show ve tahtaya yansıtmak için oluşturulan perdeyle öğrenci derse aktif olarak katılacak ve sürekli öğrencinin merakı oluşturulacaktır. Mikrofonla bilgisayara bir şeyler yazdırmak öğrencide ayrı bir heyecan oluşturacak sanki oyun oynuyormuş hissiyle daha keyifli bir

öğrenme gerçekleştirecektir. Özellikle KTM yazılımın uygulaması perdeye yansıtıldıkça öğrenci kendini televizyon veya sinema seyrediyormuş gibi mutlu hissedecektir. Böyle bir ekipman içinde öğrenciye bilgisayardaki bu programı öğretmen yardımıyla kullanabilme (fiş listesine kelime ekleme ve mikrofona bu kelimeyi okuyarak yazdırma) imkanı verildiğinde öğrenci bağımsız olarak bir şeyler yapmanın keyfini de yaşayacaktır. Bu ise öğrenciye cesaret veren bir özelliktir. Öğrenme öğretme etkinliklerinde görsel ve işitsel bazı (KTM gibi) materyallerin kullanılması oldukça önemlidir. Bu şekilde çalışma öğrencinin öğrenmesini pekiştirecek ve kalıcı bir öğrenme sağlayacaktır ve öğrencinin olumlu bir tutum kazanması mümkün olacaktır. Öğretmen açısından ise; KTM sayesinde öğretmen isterse tahtayı ve tebeşiri kullanmadan ilkokuma yazma dönemini geçirebilir. Fişleri her defasında tahtaya asmak yerine ve bu fişleri kesmek yerine programı kullanmayı tercih edebilir. Bu durumda öğretmeni bazı yüklerden arındırmış olmaktadır. Yine öğretmen de modern bir eğitim sistemi içerisinde bulunmuş olur. Teknolojiyle iç içe bir eğitim öğretim süreci gerçekleştirir. KTM' yle öğrencilere ilkokuma yazma öğretimini daha kolay sağlama imkanı bulur. Henüz ilköğretim birinci sınıfta başlayan bilgisayar kullanabilme yetisi öğrenciyi yaşam boyunca kullanacağı bir teknolojiyle iç içe bırakır. Zaten bu teknolojiyi öğrenmek ve kullanmak günümüzde neredeyse bir gereklilik halini almaktadır. Artık bilgisayarın girmediği ev sayısı azalmaktadır.

Konuşma tanıma sistemleri incelendiğinde, gerçekleştirilecek olan sistemin konuşmacıdan bağımsız hale getirilmesinde, kişiden kişiye seslerde ortaya çıkan farklılıkların hazırlanan konuşma tanıma sisteminin başarısına etki eden en önemli etkenlerden biri olduğunu söylemiştik. Konuşma tanıma başarısının yüksek olması konuşma tanıma çalışmalarında kullanılan tekniklerle paralellik sağlamaktadır. Bu amaç doğrultusunda HTK sistemi, konuşma tanıma işlemleri için gerekli olabilecek veri hazırlama, eğitim, test etme, analiz ve değerlendirme gibi modüllerden oluşmakta ve HTK ile her türlü akustik model eğitilebilmektedir. Üstelik açık kaynak koduna sahip olan bir yapıya sahiptir. Yapılan araştırmalar neticesinde HTK' nın konuşma tanıma sistemlerinde çok sıklıkla kullanıldığı görülmüştür. Bunun sebebi ise

bu sistemle çalışmanın oldukça başarılı sonuçlar vermesinden kaynaklanmaktadır. Bütün bu sebeplerden dolayı HTK kullanımı tercih edilmiştir.

Konuşma tanıma sistemlerinin temelini istatistiksel metodlar oluşturmaktadır. Bu sistemler de, tanıma işlemini gerçekleştirmek için bir eğitim aşamasına gerek vardır. Bu eğitim için kullanılan veri (ses veri tabanı) hedefte tanınmak istenen uzayı (mümkün olan bütün kelimelerin kümesi) ne kadar iyi tanımlayabiliyorsa tanıma sistemi de o kadar başarılı olmaktadır. Bu nedenden dolayı ses veri tabanı çok iyi hazırlanmalıdır.

Konuşma tanıma bilgisayar tabanlı makine ve aygıtların, ses kontrolü ile yönetilmesinin yararları insan makine ilişkili görevlerin artmasını ve daha verimli hale gelmesini sağlayabilir. Buna ilave olarak, konuşma tanımanın telefon işlemiyle ilgili uygulamalarında kullanıldığı alanlar vardır. Konuşma tanımanın kullanıldığı diğer bir alan ise, askeri ve güvenlikle ilgili alanlardır.

Bunların yanı sıra konuşma tanımanın kullanıldığı başka alanlar da vardır. İnsan-makine ilişkisi açısından, bu teknoloji özürlüler için yeni bir özgürlük sağlayacaktır. Bu teknolojinin insanlığa hizmet için bir başka kullanım alanı da; fiziksel ve görme özürlü insanlar içindir. Konuşma tanıma teknolojisi sayesinde bu insanlar yeni bir özgürlük alanına sahip olabilecektirler.

Çok fazla bilgisayara yazı yazdırma işlemlerinde, konuşma tanıma belirgin bir şekilde çalışma sürecini hızlandırabilir. Aletlerde ve makinede düzenleme yapmaktan gözleri ve elleri yorulan teknisyenler için veya bilgisayar kontrollü uygulamalarda bilgi girmek zorunda olanlar için her ikisini de yapabilme konuşma tanıma sayesinde mümkün olacaktır.

Bu olasılıklar da düşünülerek KTM v.1.1 (Konuşma Tanıyan Metin Editörü) yazılımı geliştirilmiştir. Bilgisayara bağlı bir mikrofon aracılığı ile kullanıcının konuşmasını tanıyarak ekrana yazan veya sesli komut bazında programı kontrol eden aynı zamanda Windows altında çalışan WordPad gibi bir metin

editörü şeklinde işlev gören KTM v1.1; Borland C++ Builder programlama ortamı içinde C++ dili ile geliştirilmiştir. Rich Text Format adı verilen dosya formatında çalışmak üzere tasarlanmıştır.

KTM v.1.1. yazılımında fiş listesine ilk olarak 60 cümle eklenmişti. Amaç sadece ilköğretim 1. sınıf öğrencilerine yönelik öğretilen fişlere ait cümlelerin program tarafından tanınacak cümleler olmasıydı. Bu nedenden dolayıdır ki öncelikle tanınması istenen ilk 60 kelime veya cümlelerin içeriği ilköğretim 1. sınıfta öğrencilere öğretilen fişlerin listesiydi. Yine bu sebepten yazılım sözlüğü, yazılım içerisinde fiş listesi adlı alt menü olarak kullanılmıştır. Ancak KTM içindeki fiş listesinin değiştirilebilir-düzenlenebilir özellik taşımasından dolayı amacın oldukça dışına da çıkmak mümkün olmaktadır. Aslında fiş listesi yazılımın sözlüğünü oluşturmaktadır. Fiş listesine 10.000 farklı cümle eklenebilmektedir. Bu da 10.000 farklı cümlenin KTM tarafından tanınıyor olması anlamına gelmektedir. Bu rakam sistem programlanırken sınır 10.000 alındığından dolayı 10.000 farklı cümleden bahsedilmektedir. Bu sınırın daha da yukarıya çekilmesi mümkündür. Konuşmacıdan bağımsız sürekli konuşma tanımayı sağlayan KTM v1.1, EK-2 de de gösterilen komutları sesli olarak tanımaktadır. Fiş listesindeki cümle sayısını artırarak KTM v1.1 yazılımının, konuşma tanımadaki başarısının nasıl olacağı merak edilmiştir. Bu maksatla programdaki fiş listesine, 60 cümle yanında 100 cümle daha eklenmiştir (bu cümleler “GÖÇEBE” adlı romandan alınmıştır). KTM’ ye mikrofonla önce yeni 100 cümle okunmuş, sonra 160 cümle okunmuş ve yazılımın konuşma tanıma başarısı 1. etapta test edilmiştir. Bu test etme aşamasında ve bu yazılımla çalışma aşamasında gürültü seviyesinin çok fazla olmaması gerekmektedir. Ayrıca herhangi bir gürültünün olmaması daha da iyi bir sonuç verecektir. Test sonuçları konuşma tanımada başarı sağlandığını göstermiştir. Fiş listesinde 160 cümleyle yazılımı test etmenin yetersiz olacağı düşünülerek 2. etap olarak 100 cümle daha yazılımın fiş listesine eklenmiştir. Hem yeni oluşturulan cümlelerle hem de var olan önceki cümlelerle (toplam 260 cümle) KTM v1.1 yazılımının konuşma tanıma oranı tekrar test edilmiş ve yine tanıma başarısının oldukça iyi sonuçlar verdiği gözlemlenmiştir. Farklı konuşmacılarla yeniden test edilen yazılım başarısı

yine aynı oranda başarı göstermiştir. Yine cümle sayısının yeterli olmadığı düşünülerek farklı 100 cümle daha yazılımın fiş listesine eklenerek (toplam 360 cümle) yine 3. etap olarak yazılım başarısı test edilmeye çalışılmıştır. Gözlem sonuçları yine aynı çıkmış ve yazılımın başarı oranı değişmemiştir. 4. etapta 100 farklı cümle daha yazılımın fiş listesine eklenerek yeniden yazılımın konuşma tanıma başarısı test edilmeye çalışılmıştır. Elde edilen sonuç öncekilerden farklı değildir. Yani 4. etapta da (toplam 460 cümle) KTM v1.1 konuşma tanıma başarısı oldukça yüksektir. Yine bu eklenen yeni cümleler farklı konuşmacılarla tekrar test edilmiş ve yine aynı başarı oranı gözlenmiştir. Oluşan bazı hatalar ise fiş tanıma düğmesine zamanında basılmadığı veya zamanından önce fiş tanıma düğmesinden uzaklaşıldığı veya mikrofona duyarlılık olmaktadır.

Öncelikle şunu belirtmek gerekir ki, KTM v1.1; yukarıda da belirtildiği üzere konuşma tanıma olgusunu başarıyla gerçekleştiren bir yazılımdır. Ayrıca böyle bir yazılımla, bir öğretmen öğrencilerinin dikkatini daha fazla derse çekmeyi başarabilir. Mikrofon kullanarak çalışan bir yazılım olmasından dolayı, öğrencilerin ilgisi yüksek olacaktır. Maddeler halinde anlatılması gereken konular veya konu özetleri için, öğretmenin tahtayı ve tebeşiri kullanmak istemediği zamanlar veya sürekli tahtaya bir şeyler yazmak istediği ve bunları öğrencilere dikte ettirmek istediği zamanlar, yorgun olduğu zamanlar bu yazılımı oturarak ve de dinlenerek kullanması mümkündür. Böylelikle hem öğretmene hem de öğrencilere kolaylık sağlanmış olmaktadır. Çünkü konuyu takipte geciken öğrenci, bir projeksiyon veya data show aracılığıyla tahtaya yansıtılan bilgisayar ekranında konuyu eksik kaldığı yerden takip edebilme imkanı bulmaktadır. Bu yazılım sadece metin olarak ele alındığında bir sunum programı olarak da değerlendirilebilir. Programın yazı tipi ayarlarını değiştirmek mümkün olmaktadır. Böylelikle öğrencilerin tahtayı görebileceği büyüklükte yazı boyutu ayarlanabilmektedir. Bu durum sadece öğrencilere göre düşünülmemelidir. Bir seminer esnasında veya bir konferans esnasında da kullanma olasılığı olabilen bir yazılımdır. Bunun için kablosuz fare kullanımı ve mikrofon kablosunun uzun olması gerekecektir. Ancak büyük konferans salonlarında veya çok kalabalık ortamlarda gürültü

seviyesi çok yüksek olursa programın başarımının düşmesi söz konusudur. Bütün bu sayılanların yapılması için KTM ‘yi kullanan kişinin yazılımın fiş listesi kısmını kendi çalışmalarına göre düzenlemesi gerekmektedir. Takdir edilmelidir ki on binlerce sayıda farklı cümle oluşturabilmek mümkündür. Birbirinden farklı cümle oluşturma olasılığının bu denli çok çok fazla olması; fiş listesinin sunumdan önce düzenleme zorunluluğunu doğurmuştur. KTM’ nin en önemli özelliği fiş listesinin anlatımı veya sunumu yapılacak içeriğe göre değiştirilebilir ve düzenlenebilir olmasıdır. Bu içerik tek bir sefere mal olmak üzere fiş listesine eklenerek istenildiği kadar sayıda çalışma ortamında kullanılabilir. Örneğin üç farklı ders için üç farklı içerik fiş listesine eklenebilir. Yukarıda da belirtildiği gibi fiş listesine 10.000 farklı cümle eklemek mümkündür. Diğer taraftan ise KTM yazılımı tıp alanında da kullanılabilir. Konuşma terapisi için çok ideal bir yazılımdır.

KTM’ nin olumsuz yönü ise; komutları veya cümleleri programdan tanımasını isterken, fareyle komut tanıma veya fiş tanıma düğmesi üzerinde basılı tutma zorunluluğunun olmasıdır. Ama KTM’ nin online olarak çalışılması zorunluluğu düşünülmüş ve ne zaman tanıma işlemine başlayacağının belirlenmesi için de düğme-fare özelliği kullanılmıştır.

KTM programının bir diğer özelliği ise tanınması istenilen cümleye veya kelimeye yakın bir sürekli konuşma tanıma yapabilir olmasıdır. Örneğin fiş listesine “O her gece üstündekilerle yatıyor” şeklinde bir cümle eklenmiş olsun ve bu cümleyi söylerken ya da konuşurken “her gece üstündekilerle yatıyor” olarak eksik söylendiği varsayılsın. Tanıma işlemi gerçekleştiğinde ekrandaki ifadenin “O her gece üstündekilerle yatıyor” şeklinde fiş listesinde yazıldığı gibi tanındığı görülecektir. Bir başka örnek verilecek olursa; yine fiş listesine “bir hafta yedi gündür” şeklinde bir cümle eklenmiş olsun. Sonra bu cümle “bir haftada yedi gündür” olarak değişik veya hatalı bir şekilde mikrofonla seslendirilerek KTM‘ nin bunu tanıması düşünülsün. Uygulandığında ekranda yazılan cümlenin “bir hafta yedi gündür” olarak tanındığı görülür. Özetle fiş listesine bir cümle nasıl yazıldıysa noktalama işaretleri dahil olmak üzere tanıma işlemi esnasında da aynen tanıma yapılır.

Zaten doğrusu fiş listesinde yazılı olan cümlelerin veya kelimelerin eksikliğinin, mikrofonla konuşurken oluşabilecek herhangi bir heyecan veya ufacık hatadan dolayı ortaya çıkacağı düşünülerek bu durum bu şekilde düzenlenmeye çalışılmıştır. Word ortamında hatalı yazılan bir kelimeye, programın o kelimeye yakın bir başka kelime önerebiliyor olması düşüncesi bizi bu şekilde düşünmeye zorlamıştır.

Konuşma Tanıma sistemleri dilden bağımsız olarak hazırlanabilmelidir. Bu amaç doğrultusunda yeni algoritmalar geliştirilmelidir. Ülkemizde konuşma tanıma çalışmalarında yapay zeka tekniklerinden yapay sinir ağlarına ait çeşitli algoritmalar üzerinde çalışılmıştır. SMM kadar çok sıklıkla kullanılmasa da yapay sinir ağlarıyla ilgili yapılan çalışmalarda, ses eğitme aşamasının oldukça uzun sürdüğü yapılan araştırmalarda görülmüştür. Kelime veya cümle sayısının artmasıyla bu başarı oranı daha da düşmektedir. Diğer ülkelere bakıldığında konuşma tanıma sitemlerinde genel olarak SMM yapısı kullanılmıştır. Bunun yanı sıra diğer yapay zeka teknikleriyle de çalışılmıştır (Genetik algoritma, fuzzy logic, uzman sistemler). Öyle ki hybrid sistemler denilen karma veya melez sistemlerle bu çalışmalara ağırlık verilebilir. Hem yapay sinir ağı ve hem SMM sistemleri birlikte kullanılabilir. Bu örnek diğer yapay zeka teknikleriyle de çoğaltılabilir. Ülkemizde konuşma tanıma teknolojilerinde yapay sinir ağları haricinde diğer yapay zeka teknikleriyle çalışılmadığı gözlenmiştir. Ayrıca konuşma tanıma çalışmaları için hazır ses veri tabanı kütüphanesi oluşturulmalıdır. Tubitak bu konuyla ilgili olarak çalışmalarına devam etmektedir. Türkçe konuşma tanıma yapabilen sistemlerin geliştirilmesi üzerine daha çok araştırma yapılmalıdır. Özellikle konuşmacıdan bağımsız bağlı konuşma tanımayı sağlayan çalışmalar daha kullanışlı olacaktır. Örneğin mahkeme duruşmalarında, emniyet sorgularında, zabıt işlemlerinde hep karşılıklı konuşmaların anında bilgisayara yazılması söz konudur. Böyle bir alana yönelik yapılan çalışma oldukça kullanışlı olacaktır. İnsanlığı daha rahat bir çalışma ortamına kavuşturabilmek için konuşma tanımayla ilgili çalışılabilecek alanların çeşitliliğinin artırılması gerekmektedir.

KAYNAKLAR

1. Acar, D., "Triphone Based Turkish Word Spotting System", Master Thessis, ***The Department of Electrical And Electronics Engineering, The Middle East Technical University***, Ankara, 68 (2001).

2. Akçay, B., "Yapay Sinir Ağları İle Türkçe Konuşma Tanıma", Yüksek Mühendislik Tezi, ***Hacettepe Üniversitesi Fen Bilimleri Enstitüsü***, 1-10 Ankara, (2004).

3. Akçayol, M.A., "Bir Anahtarlamalı Relüktans Motorun Sinirsel-Bulanık Denetimi", Doktora Tezi, ***Gazi Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, (2001).

4. Akpınar, H., "Yapay Sinir Ağları ve Kredi Taleplerinin Değerlendirilmesinde Bir Uygulama Önerisi", ***İstanbul Üniversitesi İşletme Fakültesi***, İstanbul, 4-7 (1993).

5. Albesano, D., Gemello, R., and Mana, F., "Hybrid HMM-NN Modeling of Stationary-Transitional Units for Continuous Speech Recognition", ***Elseiver Information Sciences***, 3 (2000).

6. Anderson, W. H., "Speech Recognition Finds Its Voice, Helping Lawyers Leverage Technology", ***Toronto LegalTech.***, 3:11 (2001).

7. Artuner, H., "Bir Türkçe Fonem Kümeleme Sistemi Tasarımı ve Gerçekleştirimi", Doktora Tezi, ***Hacettepe Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, 47-55 (1994).

8. Bayri A. ve Bingöl S., "Sözcük-altı Modeller Kullanarak Ayrık Sözcük Tanıma", ***SİU' 98 6. Sinyal İşleme Uygulamaları Bildirimler Kitabı***, Kızılcahamam, Ankara, 2:502-503 (1998).

9. Becchetti, C., and Ricotti L. P., "Speech Recognition Theory and C++ Implementation", ISBN 0-471-97730-6, ***John Wiley & Sons Ltd,*** England, 167-188, 310-311 (1999).

10. Cansız, M., "Yapay Sinir Ağları İle Kişilerin Ses Örneklerinden Kimliklerinin Tanınması", Yüksek Lisans Tezi, ***Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü***, Afyon, 1 (1997).

11. Charniak, E., and Mcdermott, D., "Introduction to Artificial Intelligence", ***Addison Wesley Longman Publishing Company***, ISBN 0-201-11946-3, Boston, USA, (1985).

12. Chassaing, R., "Digital Signal Processing with C and the TMS320C30", ***A Wiley-Interscience Publication***, ISBN 0-471-55780-3, USA, 324-325 (1992).

13. Çilingir, O., "Large Vocabulary Speech Recognition for Turkish", Master Thessis, ***The Department of Electrical And Electronics Engineering, 9The Middle East Technical University***, Ankara, (2003).

14. İnternet: Tubitak-Turtel Ses Tabanının Ayrık Kelime Tanıma Başarımının Konuşmacı Bazlı Değerlendirilmesi, http://www.siu2002.isikun.edu.tr/html/iu2002_full%5Csiu2002_no214.pdf (1997).

15. Doğan, S., "PC Ortamında Sesli Komutları Tanıma", Yüksek Lisans Tezi, ***Marmara Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 4,15-27 (1999).

16. Durmuş, E., ve Gül E., "Yazıyı Sese Çevirim", ***SİU' 98 6. Sinyal İşleme Uygulamaları Bildirimler Kitabı***, Kızılcahamam, Ankara,1:233 (1998).

17. Furui, S., "Digital Speech Processing", Synthesis and Recognition", ***Marcel Dekker Inc***., New York, (1989).

18. Gökhan, A., "Yapay Sinir Ağları İle Ayrık Türkçe Sözcüklerin Tanınması", Yüksek Lisans Tezi, ***Fırat Üniversitesi Fen Bilimleri Enstitüsü***, Elazığ, 1-17 (1997).

19. Jelinek, F., "Statistical Methods for Speech Recognition", ISBN 0-262-1006-5, ***The MIT Preess Cambridge***, Massachusetts London, England, 14-18, 15-27 (1997).

20. Jurafsky, D. & Martin, J. H., "Speech and Language Processing An Inroduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", ISBN 0-13-122798-X, ***Prentice Hall Upper Saddle River***, New Jersey, USA, 235-249 (2000).

21. Kabre, H. and Spalanzani, A., "EVERA: An Evolutionary Programming Environment for Adaptive Speech Processing, Information Sciences", ***Elsevier Science Inc***., USA,122(2-4):185-200 (1999).

22. Kılıç, Ç. "İlkokuma Yazma Yönteminde Kullanılan Yöntemler", Eğitimde Reform Dersi Tezsiz Yüksek Lisans Programı, ***Ankara Üniversitesi Eğitim Bilimleri Enstitüsü*****,** Ankara, 6 (2005).

23. Lippmann, R.P., "An Introduction to Computing with Neural Nets." ***IEEE Acoustics, Speech and Signal Processing Magazine***, 4:4-22 (1987).

24. Maren, A. et al, "Handbook of Neural Computing Applications", ***Academic Press Inc***., ISBN 0-12-4711260-6, Orlando, USA, (1990).

25. Massaro, D. W. et al, "Picture My Voice: Audio to Visual Speech Synthesis Using Artificial Neural Networks", ***Perceptual Science Laboratory***, University of California, August, U.S.A, 2-3 (1999).

26. Mengüşoğlu, E., "Bir Türkçe Sesli İfade Tanıma Sisteminin Kural Tabanlı Tasarımı ve Gerçekleştirimi", Yüksek Mühendislik Tezi, ***Hacettepe Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, 14-16, 22-26 (1999).

27. Morgan, D. And Scofield, L. C., "Neural Networks and Speech Processing", ***Kluwer Academic Publishers***, USA, 102-108 (1991).

28. Niesler, T., "Category-based Statistical Language Models", P.h.D. Thesis, ***Cambridge University***, June, Cambridge, 11-12 (1997).

29. Öncül, N., "Kısa Eğitme Süreli Bir Konuşmacı Tanıma Dizgesi Tasarımı ve Gerçekleştirilmesi", Yüksek Mühendislik Tezi, ***Hacettepe Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, (1993).

30. Rabiner, L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", ***Proceedins of IEEE***, 77(2):257-286 (1989).

31. Rumelhart, D.E. ve McClelland, J.L, "Parallel Distributed Processing", ***MIT Press***, July, Cambridge, 1:10-16(1986).

32. Schroeder, M. R., "Computer Speech Recognition, Compression, Synthesis", ISBN 3-540-21267-1, ***Printed in Germany,*** Springer, 55-59, 166 (2004).

33. İnternet: "İlköğretimde Etkili Öğretme ve Öğrenme Öğretmen El Kitabı", www.idilyibo.com/DERS%20NOTLARI_dosyalar/Modul-5.pdf, (1999).

34. Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., and etc. , "The HTK Book (for HTK Version 3.1)", ***Copyright (1995-1999) Microsoft Corparation, Copyright (2001-2002), Cambridge University Engineering Department***, Cambridge, 3-19 (2002).

EKLER

EK–1 Ses veri tabanında kullanılan bazı cümleler

Ses veri tabanında kullanılan 1600 cümleden 150 tanesi örnek oluşturması açısından aşağıda listelenmiştir.

o , bütün yıl , senin elbiseni yağlı yıkama suyunda tuttu
bana yağlı bir paçavrayı bu şekilde tutmamı söyleme
bu bizim için kolaydı
jale çok sıkı çalışarak , daha fazla para kazanabilir
o benden daha zayıf
parlak güneş ışığı okyanus üzerinde parıldıyor
hiçbir şey suçsuzluk kadar hatır kırıcı değildir
aptal maddeler üzerinde üzülmek ya da avazı çıktığınca bağırmak niye
biz uzaktayken sen neredeydin
senin notların naciye'ninkilerden düşük mü yüksek mi
o , az bulunur bir yalana izin verecek
rana sarı bir zambak giyecek mi
kollarını olabildiğince yükseğe salla
perşembe günkü sınavdan önce , her formülü tekrar et
müze her akşam değişik müzisyenler kiralıyor
duvarın kenarında bir tomar tel duruyor
kerime neşeli bir evde yaşıyor
nafaka boşanmış bir adamın zenginliğine zarar verir
alüminyum gümüş eşyalar genellikle film gibi ince olabilir
o , sıcak , kürklü , yünlü bir tulum giymişti
alfa alfa sizin için sağlıklıdır
başka hiç bir şey işe yaramadığı zaman kuvvet kullan
Su müzisyenler , olağanüstü bir şekilde birbirlerime uyum sağlıyorlar
her zaman yalnız olmamıza rağmen hayatta kalırız
sadece avukatlar milyonerleri sever
gençlerin çoğu sabah erken kalkar
babam akademik açık arttırma yaptı mı
o nöbetçiye bir litre gaz için yalvar

EK–1 (Devam) Ses veri tabanında kullanılan bazı cümleler

gamzeye bir çuval patates toplaması için yardım et
bakmak için bir tekir kedi edin
iyi bir davranış yenilmezdir
bataklık kuşlarını odun ateşinde kızartmak çok eğlencelidir
hindistan cevizi kremalı kek çok güzel bir tatlıdır
ahmedin kirli tabaklarını yıkama
erkek kardeşinin başarısını kutlamaya yardım et
sadece , en ustalaşmış sanatçılar üne sahip olur
önemli donanım uygun bakma gereksinim duyar
genç insanlar atletik müsabakalara katılırlar
banunun altın bileziği mezuniyet hediyesiydi
tartışmaları kışkırtmak öğrencilerin dikkatini sürekli kılıyor
görgü kuralları varolan kurallarla uyumluluğu buyurur
din öğrencileri tarihi tartışıyorlar
yaşlı insanlar çoğunlukla suçlanırlar
basketbol eğlendirici bir spor haline getirilebilir
toplama ve çıkartma öğrenilen yeteneklerdir
yankesici suçüstü yakalandı
annem aperatifleri hiç sevmez
anneannem modernlik uğruna iç eteklik giymekten vazgeçti
on ikinci günde alacakaranlıkta osmana kavuşacağız
ağır ekonomik kesintiler fakirleri hiç göz önünde tutmuyor
iki elini de kullanabilen yankesiciler daha başarılı oluyor
onun klasik yapıtı , eleştirmenlerin beğenisini kazandı
çok küçük bir sözlük bile semboller içerir
doğu sahili saf mutluluğun ve heyecanın tam yeridir
ısıtmanın yetersizliği kiracının dert yanmasına sebep oldu
akademik yetenek ve eğilim sana diplomayı garanti eder
avcı, görüntüsünü değiştirmek için kayak maskesi takmıştı
derece yaparken sıkıntıya ve engellemelere maruz kalıyoruz
şarkıcının parmağına kıymık batmıştı

EK–1 (Devam) Ses veri tabanında kullanılan bazı cümleler

kanun yapıcı kuralların hepsi , kamu eğitimini yargılamak zorunda kaldı
çikolata ve güller her zaman romantik hediyeler olarak kalacaklar
herhangi bir katılım çok iyi bir şekilde ödüllendirilecektir
kıtaların sürüklenmesi coğrafi bir teoridir
düzenli olarak devam etmek pek aranmaz
her generalin zekasına meydan okudu
hiç durmadan yağan yağmurda iliklerimize kadar ıslandık
geçen yıl görülen gaz sıkıntısı aşırı fiyat yükselmesine neden oldu
zenginliğini başkalarına yansıtmak için statünü iyileştir
kuru üzümlerini dışarıda , kapı saçaklığı basamaklarında ye
daha yemedin mi
oklu kirpi deniz kestanesine benzer
bahar caddesine buradan dümdüz gidebilirsiniz
özgenin gösterisi ekrana yanlış yerleştirilmiş
resmi bir songün ertelenemez
o kantini taze kaynak suyuyla doldur
melikenin köpük heykelini , nazikçe incitmeden kutuya yerleştiriniz
gayda ve davul müzik aletleridir
doktorlar özgürce ilaç reçetesi verebiliyorlar
Lütfen , oluşan budalaca durumu anlatır misiniz
bir çeki ertelemek kanunlara aykırıdır
bürokrasiyle uğraşmak imkansızdır
iyi hizmet büyük bahşişlerle ödüllendirilmelidir
benim açıklamalarımın acilen yenilenmeye ihtiyacı var
anlayışla beraber işbirliği tartışmaları en aza indirir
çimento , metreküple ölçülür
ispanyol kostümleri oldukça renkli
ilkel kabilelerin neşeli ve iyimser bir davranış şekli var
eğer tasarruf etmek istiyorsan yedek uçuşlar sana uygun olabilir
gerçeğini taklidinden ayırmak çok güç
tuncer meşhur bir heykeltraştır

EK–1 (Devam) Ses veri tabanında kullanılan bazı cümleler

yanlış basım ani bir iddia reddine sebebiyet verdi
geniş bir ev halkı , çok sayıda düzenleme gerektirir
her kenardan kendine küçük bir köşe kes
çocuklar sıradan şekerlemeleri , olağandışı bir şeymişcesine severler
iguanalar ve timsahlar tropikal sürüngenlerdir
maskeli balolar insanın hayal gücünü kamçılıyor
penguenler buzlu antartikaya yakın yaşarlar
soruyu yanıtından tahmin edeceksiniz
ortaçağa ait toplumlarda yönetim , hiyerarşiye dayalıydı
projenin gelişimi çok yavaş ilerleme kaydediyordu
anaokulu çocukları her bayram sınıflarını süslerler
özel görevli kuvvetler misafirleri korsanlardan korurlar
tıbbi yardım için cankurtaran çağırın
o bir dilenciden onbin lira çalmış
dosyaları kesinlikle silmelisiniz
evinin girişindeki duvarda , büyük bir kaneviçe işi asılıydı
doğum günlerinde dondurma ve pasta olur
başının üstü bugünkü güneş yüzünden su topladı
dik yamaçta ayağı kaydı ve ayak bileği burkuldu
öğrenmenin en iyi yolu fazladan soru çözmektir
onun ani gidişi ekibi şaşkına uğrattı
römorkörler devasa yükleri çekme kapasitesine sahiptirler
kaslı bir karın sırtın için çok faydalı bir şey
karikatür bir iribaşı ve bir köstebeği resmetmiş
amblem akrepolisi tamamen ateş içinde gösteriyor
vidayı sol elinle sık
mango ve papaya kasenin içinde duruyor
bileşimdeki maddelerin hepsini büyük bir kasede karıştır
yanlış bildirim özür dilenerek geri alındı
kurt kaplan ve hiyena vahşi hayvanlardır
kanuna karşı gelmek yasak ve cezaya tabidir

EK–1 (Devam) Ses veri tabanında kullanılan bazı cümleler

ansiklopediler hemen hemen hiç öykü türü şeylere yer vermezler
tornavida , votkadan ve portakal suyundan yapılmıştır
kalaylı alaşımdan yapılan eşyalar güzel oluyor
güzelyalı çanakkalenin bir sayfiye yeridir
yapay zeka bir gün gerçek olacak
imparator iyi huylu bir adamdı
çok sayıda yabancı filmin altyazısı var
ankara kedileri siyam kedilerinden daha tüylüdür
sabah kahvaltısında dört tane fazla yumurta yedi
biz yeni bir içecek fabrikası kurmayı planlıyoruz
teşhircilik ve dile düşmüşlük elele giderler
pizzalar hızlı bir öğle yemeği için büyük kolaylık
aralık ve ocak mısırda geçirilecek en güzel aylar
teknik yazarlar bibliyografya yazarken kısaltmalar kullanabilirler
bilimsel ilerleme yeni tekniklerin oluşturulmasından ileri gelir
geleneklere göre , küçük yaşta evlilik için ailelerin onayı gerekir
sakar müşteri pahalı parfümden biraz döktü
bungalov mutlu bir şekilde deniz kıyısına yerleştirilmiş
tarım ürünleri düzensiz bir şekilde dağıtılmış
antalyanın su yarışlarına katılmaya söz ver
hangi uzun makale yeterince açık değildi , ve netleştirilmesi gerekiyordu
aylinin borazanının sesi antilobu ürküttü
fidan gibi ince kadın kürk giymişti
fazla merak başını belaya sokabilir
dönüp dolaşan programlar asla derlenmez
talimatlarımın doğru yerine getirilmesi çok mühim
dairelerin çoğunda oyların üçte biri sayıldı
tülayı beklerken , meydanda defalarca volta attı
vietnam mutfağı enfestir

EK–2 Programı kontrol eden komut listesi

Programı kontrol eden komut listesi

yeni
aç
kapat
kaydet
farklı kaydet
yazdır
Ön izleme
programdan çık
geri al
kes
kopyala
yapıştır
tümünü seç
yazı tipi
büyük küçük harf
tarih saat ekle
metin ara
sonrakini bul
değiştir
araç çubuğu
durum çubuğu
ses aygıtı
cetvel
komut tanıma
fiş tanıma
fiş listesi
ardışıl sırala
yatay sırala
dikey sırala
hepsini küçült

EK–2 (Devam) Programı kontrol eden komut listesi

konular
hakkında

ÖZGEÇMİŞ

Nursel Yalçın, Kayseri'nin İncesu ilçesinde doğdu. İlk ve orta öğrenimini burada tamamladı. 1994 yılında Gazi Üniversitesi Endüstriyel Sanatlar Eğitim Fakültesi Bilgisayar Eğitimi Bölümünden fakülte birincisi olarak mezun oldu. 1994 Haziranında da aynı fakültenin Bilgisayar Eğitimi Bölümüne araştırma görevlisi olarak atandı. Aynı bölümde "Yazılım Geliştirme Sürecinde Yazılım Kalite Standartları" konulu çalışmasıyla yüksek lisans derecesini aldı. 1997 yılında bir yıl süreyle Orta Doğu Teknik Üniversitesi Yabancı Diller Yüksek Okulunda Temel İngilizce hazırlık eğitimi gördü. Almanca da bilen Yalçın, 2000 yılında Fen Bilimleri Enstitüsünün Endüstriyel Teknoloji Eğitimi Ana Bilim Dalında doktora programına başladı.