TÜRKÇE KONUŞMA TANIMA SİSTEMLERİ İÇİN BİR KONUŞMA VERİTABANI

Barış Safa GÜRLER

YÜKSEK LİSANS TEZİ
ELEKTRONİK – BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

HAZİRAN 2014

Barış Safa GÜRLER tarafından hazırlanan "TÜRKÇE KONUŞMA TANIMA SİSTEMLERİ İÇİN BİR KONUŞMA VERİTABANI" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ / OY ÇOKLUĞU ile Gazi Üniversitesi Elektronik-Bilgisayar Eğitimi Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Nurettin DOĞAN

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum ...…….

Başkan : Prof. Dr. Şeref SAĞIROĞLU

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum ...…….

Üye : Yrd. Doç. Dr. Nursel YALÇIN

Yazılım Eğitimi Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum ...…….

Tez Savunma Tarihi:

27/06/2014

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……..…….

Doç. Dr. Nurettin TOPALOĞLU

Bilişim Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

.....

Barış Safa GÜRLER

27/06/2014

TÜRKÇE KONUŞMA TANIMA SİSTEMLERİ İÇİN BİR KONUŞMA VERİTABANI

(Yüksek Lisans Tezi)

Barış Safa GÜRLER

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Haziran 2014

ÖZET

Konuşma tanıma, söylenen sözlerin metne dönüştürülmesidir. Sesle kontrol uygulamalarının yanı sıra çeşitli dikte işlemleri için de kullanılmaktadır. Günümüzde konuşma tanıma uygulamaları daha çok Saklı Markov Modeli adı verilen yaklaşımla geliştirilmektedir. Saklı Markov Modelleri (SMM), görünür çıktılar üreten, ancak arka planda bu çıktıların üretilmesine neden olan saklı durum geçişlerini içeren durumları modellemek için geliştirilmiştir. Konuşma tanıma problemi de bu tanıma uymaktadır. Konuşma tanıma, pek çok alt problemden oluşan, karmaşık bir disiplindir. Konuşma tanıma sistemlerinin geliştirilmesi esnasında, konuşma veritabanlarının oluşturulması oldukça önemli bir husustur. Bu çalışmada, Türkçe konuşma tanıma sistemleri için bir Türkçe konuşma veritabanı geliştirilmiştir. Okuma konuşması hedef alınmıştır. Konuşma veritabanı geliştirme aşamaları adım adım incelenmiş, bu adımlarda dikkat edilmesi gereken noktalar ve kullanılan araçlar ele alınmıştır. Ses kayıtları, 1989 - 1995 arası doğumlu 30 konuşmacıya (15 erkek, 15 kadın) 60'ar cümle okutularak elde edilmiştir. Veritabanında ses kayıtlarının yanı sıra Hidden Markov Model Toolkit (HTK) ile oluşturulan fonem seviyesindeki zaman damgaları da yer almaktadır. Metin işleme ve Türkçe sözcük altı istatistik işlemleri için C# dilinde programlar yazılmıştır.

Bilim Kodu : 702.1.014
Anahtar Kelimeler : konuşma tanıma sistemleri, konuşma veritabanı, saklı markov modeli, smm, htk
Sayfa Adedi : 74
Danışman : Doç. Dr. Nurettin DOĞAN

A SPEECH DATABASE FOR TURKISH SPEECH RECOGNITION SYSTEMS

(M. Sc. Thesis)

Barış Safa GÜRLER

GAZİ UNIVERSITY

INFORMATICS INSTITUTE

June 2014

ABSTRACT

Speech recognition is translation of spoken words to text. It is used for dictation as well as voice user interfaces. Today, speech recognition systems are mostly developed with the Hidden Markov Model (HMM) approach. Hidden Markov Models are developed for modelling visible output emitting situations which contain the hidden state transitions in background that cause those outputs to be generated. Speech recognition problem fits that definition. Speech recognition is a complex discipline that consists of many sub problems. Speech database construction is a very important matter in developing speech recognitions systems. In this study, a Turkish speech database for Turkish speech recognition systems has been constructed. Reading speech has been set as target. Speech database construction stages are investigated step by step, the most sensitive spots in those steps and different tools that have been used are mentioned. Audio recordings obtained from 30 speakers (15 male, 15 female) which were born between 1989 and 1995. Phoneme level timestamps generated with Hidden Markov Model Toolkit (HTK) are in the database alongside the audio recordings. Programs are written in C# for text processing and subword statistics of Turkish language.

Science Code : 702.1.014
Key Words : speech recognition, speech database, hidden markov model, hmm, htk
Page Number : 74
Supervisor : Assoc. Prof. Dr. Nurettin DOĞAN

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren hocam Doç. Dr. Nurettin DOĞAN'a teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ viii

ŞEKİLLERİN LİSTESİ ix

RESİMLERİN LİSTESİ x

SİMGELER VE KISALTMALAR xii

1. GİRİŞ 1

2. KURAMSAL TEMELLER 5

3. KONUŞMA VERİTABANI OLUŞTURMA SÜRECİ 13

4. SONUÇ VE ÖNERİLER 61

KAYNAKLAR 63

EKLER 65

EK-1. Telaffuz sözlüğünün (Pronunciation Dictionary) hazırlanması 66

EK-2. Kelime Listesinin (Word List) oluşturulması 68

EK-3. Hmmdefs dosyasının üretilmesi 69

EK-4. Klavuz 70

ÖZGEÇMİŞ 74

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 2.1. A geçiş matrisi 7

Çizelge 2.2. B çıkış matrisi 7

Çizelge 2.3. Π başlangıç vektörü 8

Çizelge 3.1. METUBET-IPA fonem tablosu 16

Çizelge 3.2. OTD kaynak dağılımı 20

Çizelge 3.3. Örnek cümlenin üçlüses karşılığı 25

Çizelge 3.4. Üçlüses istatistikleri 26

Çizelge 3.5. En yüksek skor alan 35 cümle 27

Çizelge 3.6. Seçilen 60 cümlenin üçlüses frekansları 37

Çizelge 3.7. Üçlüses frekans sıralama karşılaştırması 38

Çizelge 3.8. Üçlüses frekans dağılımı 39

Çizelge 3.9. En yüksek skor alan 40 kelime 40

Çizelge 3.10. Konuşmacıların memleketlerine göre dağılımı 59

Çizelge 3.11. Konuşmacıların memleketlerinin bölgelere göre dağılımı 59

Çizelge 3.12. Konuşmacıların yetiştikleri şehre göre dağılımı 60

Çizelge 3.13. Konuşmacıların yetiştikleri şehirlerin bölgelere göre dağılımı 60

Çizelge 3.14. Konuşmacıların doğum yıllarına göre dağılımı 60

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 1.1. Ses teknolojileri 1

Şekil 1.2. KT uygulama çeşitleri..... 2

Şekil 2.1. 3 durumlu Markov zinciri..... 5

Şekil 2.2. Hava durumu-eylem Markov modeli..... 6

Şekil 2.3. Olasılık hesabı (1)..... 8

Şekil 2.4. Olasılık hesabı (2)..... 9

Şekil 2.5. Olasılık hesabı (3)..... 10

Şekil 2.6. Olasılık hesabı (4)..... 10

Şekil 2.7. Olasılık hesabı (5)..... 11

Şekil 2.8. Olasılık hesabı (6)..... 11

Şekil 3.1. sp modelinin oluşturulması..... 57

RESİMLERİN LİSTESİ

Resim **Sayfa**

Resim 3.1. OTD workbench yazılımı 19

Resim 3.2. OTD dosya yapısı 21

Resim 3.3. Xcs dosyalarının içeriği 22

Resim 3.4. Düzenleme sonrası elde edilen dosya 24

Resim 3.5 (a). Sözcük altı istatistik yazılımı (1) 30

Resim 3.5 (b). Sözcük altı istatistik yazılımı (2-3) 30

Resim 3.5 (c). Sözcük altı istatistik yazılımı (4) 31

Resim 3.5 (d). Sözcük altı istatistik yazılımı (5) 31

Resim 3.5 (e). Sözcük altı istatistik yazılımı (işlem) 32

Resim 3.5 (f). Sözcük altı istatistik yazılımı (6a-6b) 32

Resim 3.5 (g). Sözcük altı istatistik yazılımı (7) 33

Resim 3.5 (h). Sözcük altı istatistik yazılımı (8) 33

Resim 3.5 (i). Sözcük altı istatistik yazılımı (9) 34

Resim 3.5 (j). Sözcük altı istatistik yazılımı (10) 34

Resim 3.5 (k). Sözcük altı istatistik yazılımı (11) 35

Resim 3.5 (l). Sözcük altı istatistik yazılımı (12) 35

Resim 3.6. Sözcük altı istatistik yazılımının çıktısı 36

Resim 3.7. Smart Voice Recorder yazılımı 41

Resim 3.8. Audacity yazılımı 42

Resim 3.9 (a). Ses dosyasındaki gürültünün giderilmesi (1) 43

Resim 3.9 (b). Ses dosyasındaki gürültünün giderilmesi (2) 43

Resim 3.10. Cümlelerin ses dosyasından ayrılması 44

Resim 3.11. Ses dosyaları 45

Resim 3.12. Derlenmiş HTK dosyaları 47

Resim **Sayfa**

Resim 3.13. Scratch dizininin yapısı 47

Resim 3.14. c01.lab dosyasının içeriği 48

Resim 3.15. Cygwin programı 48

Resim 3.16. Telaffuz sözlüğünün görünümü 49

Resim 3.17. word.list dosyasının görünümü 50

Resim 3.18. Fonem listesini içeren dosyaların görünümü 51

Resim 3.19. mlf dosyasının görünümü 51

Resim 3.20. phone0.mlf dosyasının görünümü 52

Resim 3.21. copy.scp dosyasının görünümü 54

Resim 3.22. train.scp dosyasının görünümü 55

Resim 3.23. comp.cfg dosyası 56

Resim 3.24. word1.mlf dosyası 58

SİMGELER VE KISALTMALAR

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

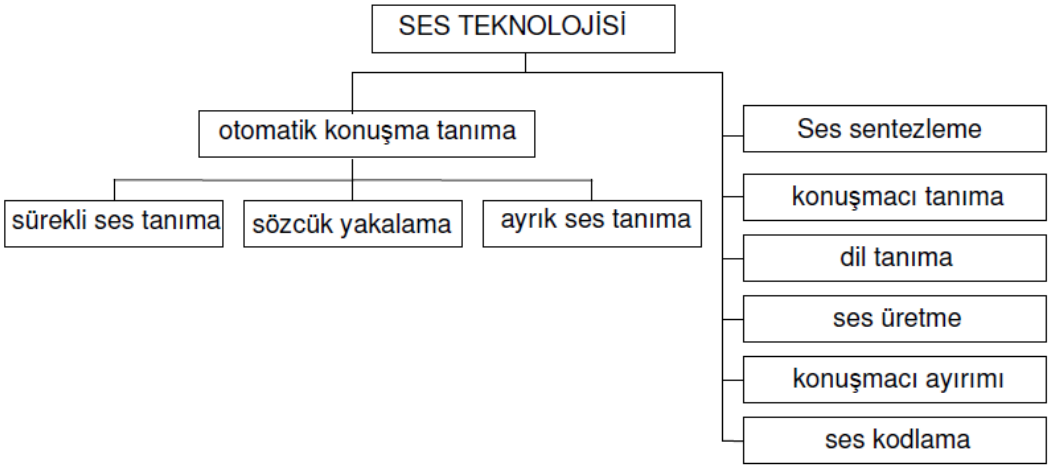
Kısaltmalar	Açıklama
ADC	Analog-Digital Converter (Analog-Sayısal Çevirici)
HTK	Hidden Markov Model Toolkit (Saklı Markov Modeli Araç Kiti)
IPA	International Phonetic Alphabet (Uluslarası Fonetik Alfabe)
KT	Konuşma Tanıma
MFCC	Mel Frequency Cepstral Coefficients (Mel-Frekansı Kepstral Katsayıları)
OTD	ODTÜ Türkçe Derlemi
SMM	Saklı Markov Modeli

1. GİRİŞ

Bu tezin amacı, Türkçe konuşma tanıma sistemlerinde kullanılabilecek bir konuşma veritabanı oluşturmaktır.

Konuşma tanıma (KT), konuşma sinyalinin bilgisayar tarafından metne dönüştürülmesi işlemidir.

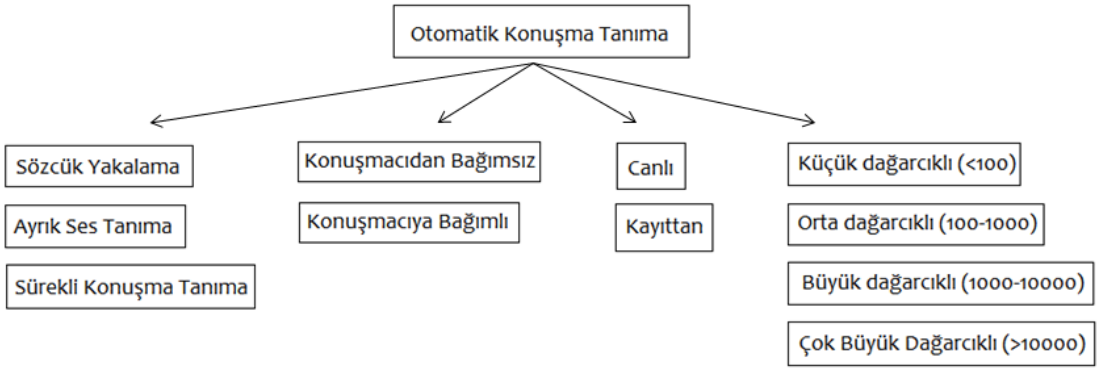
KT, ses teknolojileri (Şekil 1.1) alanında yer alan; sinyal işleme, doğal dil işleme, dinamik programlama, olasılık teorisi ve yapay zekâ gibi disiplinler tarafından desteklenen bir konudur.



Şekil 1.1. Ses teknolojileri [1]

Konuşma sinyali analiz edilerek konuşmacının kimliği tespit edilebilir (konuşmacı tanıma), konuşulan dil öğrenilebilir (dil tanıma), konuşmada geçmesi muhtemel kelimeler yakalanabilir (anahtar kelime yakalama), konuşma doğrulama yapılabilir.

Akademik ve ticari çevrelerce 1950'li yıllardan beri yürütülen KT çalışmaları, değişik tip ve çapta uygulamaların (Şekil 1.2) ortaya çıkmasını sağlamıştır.



Şekil 1.2. KT uygulama çeşitleri

KT sistemleri tanınacak ses birimine göre de sınıflandırılabilir. Genelde fonem ve sözcük tabanlı tanıma sistemleri geliştirilmişse de hece tabanlı [2] KT sistemleri de mevcuttur.

Konuşma tanıma işlemi için gereken işlemleri Yalçın [3], şöyle özetlemiştir:

"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."

KT'da karşılaşılan başlıca sorunlar şunlardır:

- Sözcük sınırları belirgin değildir.
- Fonemler, ön ve arkasına gelen ses birimleri tarafından etkilenirler.
- Hızlı konuşma söz konusu olduğu için, bazı seslerin kaybedilmesi ve/veya değişime uğraması söz konusudur.
- Aynı dili konuşan insanlar arasında bile, lehçe gibi farklılıklardan doğan farklı telaffuzlar görülmektedir.

- Aynı kişinin farklı zamanlardaki konuşmalarında bile farklılıklar gözlemlenmektedir [4].

Bu tez çalışması için yapılan araştırmada karşılaşılan konuşma veritabanları hakkında elde edilen bilgiler aşağıda sunulmuştur.

Türkçe dili için geliştirilen konuşma veritabanlarından biri ODTÜ Türkçe Mikrofon Konuşması Veritabanı'dır [5]. 193 kişiye 40'ar cümle okutularak oluşturulan bu veritabanında her ses dosyası için fonem, Saklı Markov Modeli durumu ve kelime seviyesinde hizalama dosyaları bulunmaktadır. Veritabanında ayrıca, her konuşmacı için yaş, bölge, cinsiyet, eğitim gibi bilgileri içeren birer bilgi dosyası vardır.

Bu alanda yapılan diğer bir çalışma OrienTel'dir. Akdeniz çevresindeki Kuzey Afrika ve Orta Doğu ülkelerinde konuşulan dillerde telefon hattı üzerinden konuşma veritabanlarını içeren bir projedir. 21 dil için; 1700 konuşmacının cinsiyet, yaş, şive ve arama ortamı bakımından dengeli kayıtları yapılmış, çözümlenmiş ve dökümanlaştırılmıştır. Veritabanı rakamlar, numaralar, zaman, tarih, sözcükler ve cümleler içermektedir [6].

Telefon hatları üzerinden kayıt yapılarak oluşturulan bir diğer veritabanı da özellikle adli çalışmalara destek vermek amacıyla oluşturulmuştur. 8 KHz, 8 bit'lik wav dosyalarını içeren veritabanında 55 konuşmacıdan alınan 5390 ses örneği yer almaktadır [7].

Geliştirilmiş olan Türkçe konuşma veritabanlarından biri de, Sabancı Üniversitesi öğrencileri tarafından bir proje dersi kapsamında her dönem düzenli olarak toplanan 6 yıllık bir çalışmanın ürünü olan SUVoice'dur. Toplam 3444 okuyucunun ses kayıtlarını içermektedir. Yaklaşık 70 saati sessizlik olan 185 saatlik konuşma verisi içerir. 46 farklı metin dosyasından 4087 özgün cümlenin okunmasıyla oluşturulmuştur. Veritabanı çoğunlukla 18-25 yaş grubundaki konuşmacıların verisini içerir. Konuşmacıların 2055'i erkek, 1389'u ise bayandır. Akustik modelin eğitimi için Türkçe'deki 29 harfe karşılık gelen 29 fonem kullanılmıştır. Bunlara ek olarak kelimeler arasındaki kısa boşlukları yakalamak için kısa durak ve cümle aralarındaki boşlukları yakalamak için sessizlik modelleri eğitilmiştir. Bunların yanında yabancı bazı harflerin fonemleri de önlem olarak oluşturulmuş ve toplamda 34 fonem elde edilmiştir [8].

55 adet Türkçe filmden çıkarılmış 5304 tane konuşma sinyali ve bunların metinsel içeriklerinden oluşan Türkçe Duygusal Konuşma Veritabanı'nın amacı ise Türkçe duygusal konuşmanın akustik analizi ve duygu tanıma üzerine yapılacak çalışmalara yardımcı olmaktır. Konuşma sinyallerinin hem kategorik (mutlu, şaşkın, üzgün, kızgın, korku, nötr ve diğer) hem de 3 boyutlu duygu uzayında (Değerlik, aktivasyon ve baskınlık) etiketlenmesi çok sayıda dinleyici tarafından gerçekleştirilmiştir [9].

Dünyada konuşma tanıma alanında yapılan başlıca çalışmalara değinilecektir.

1970'lerin başlarında, KT için Saklı Markov Modelleme (SMM) yaklaşımı Princeton Üniversitesi'nde Lenny Baum tarafından keşfedilmiştir. SMM, karmaşık bir matematiksel örüntü eşleme stratejisi olarak tanımlanabilir ve içinde Dragon Systems, IBM, Philips ve AT&T'nin de bulunduğu bir çok konuşma tanıma şirketi tarafından kullanılmıştır [10].

1995 yılında, ilk kez Dragon Systems tarafından üretilen kelime tabanlı dikte yazılımı piyasaya sürülmüştür. Bunun ardından, benzer yazılımlar IBM ve Kurzweil tarafından da üretilmeye başlanmıştır [10].

1996'da Charles Schwab ve Nuance tarafından Voice Broker isminde bir konuşma tanıma sistemi geliştirilmiş ve bu sistemle 360 adet müşteri telefon üzerinden aynı anda borsa işlemi yapmıştır. Bu sistem, her gün 50000 adet isteği yerine getirebilmiştir. Sistemin doğrulunun %95 civarında olduğu belirlenmiştir. Yine aynı yıl Dragon Systems "Naturally Speaking" 'i geliştirmiş ve bu ürün ilk sürekli dikte yazılımı olmuştur [10].

Günümüzde İngilizce dili için geliştirilmiş çok sayıda konuşma tanıma sistemi mevcuttur. Örnek olarak Google Voice Search, Dragon NaturallySpeaking, TalkTyper gibi uygulamalar verilebilir.

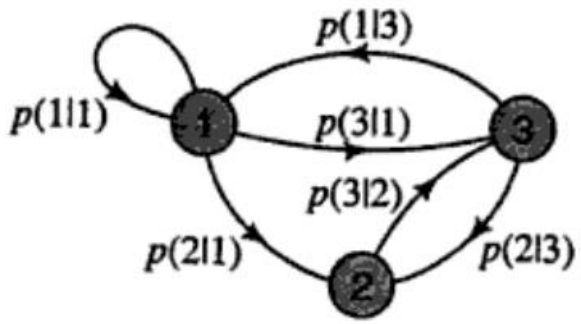
Türkçe konuşma tanıma uygulamalarında ise Sestek ve Dikte firmalarının çözümleri öne çıkmaktadır. Sestek firması aynı zamanda konuşma sentezi, ses biyometrisi gibi alanlar için de çözümler sunmaktadır. Dikte firması ise geçmişte DikteAPIF20 SDK isimli bir yazılım geliştirme kitini ücretsiz kullanıma açmıştır.

2. KURAMSAL TEMELLER

Bu bölümde, KT probleminin çözümünün temelini oluşturan SMM'lerle ilgili bilgiler verilecektir.

Markov zinciri, olasılık teorisinin çok önemli ve iyi çalışılmış bir kavramıdır. Geçmişe dair en az miktarda bilgiyi (hafıza) hesaba katan (tamamen hafızasız kalmamak kaydıyla) rassal süreçlerden oluşan bir sınıf ile ilgilenir [11].

Şekil 2.1, geçiş olasılık değerleriyle iliştirilmiş olan oklar, durumlar arasındaki geçişleri göstermektedir. Bazı geçişlerin olmaması, p(1|2) = p(2|2) = p(3|3) = 0 olduğunu ima etmektedir. Şekil, sürece neden zincir adı verildiğini göstermektedir [11].

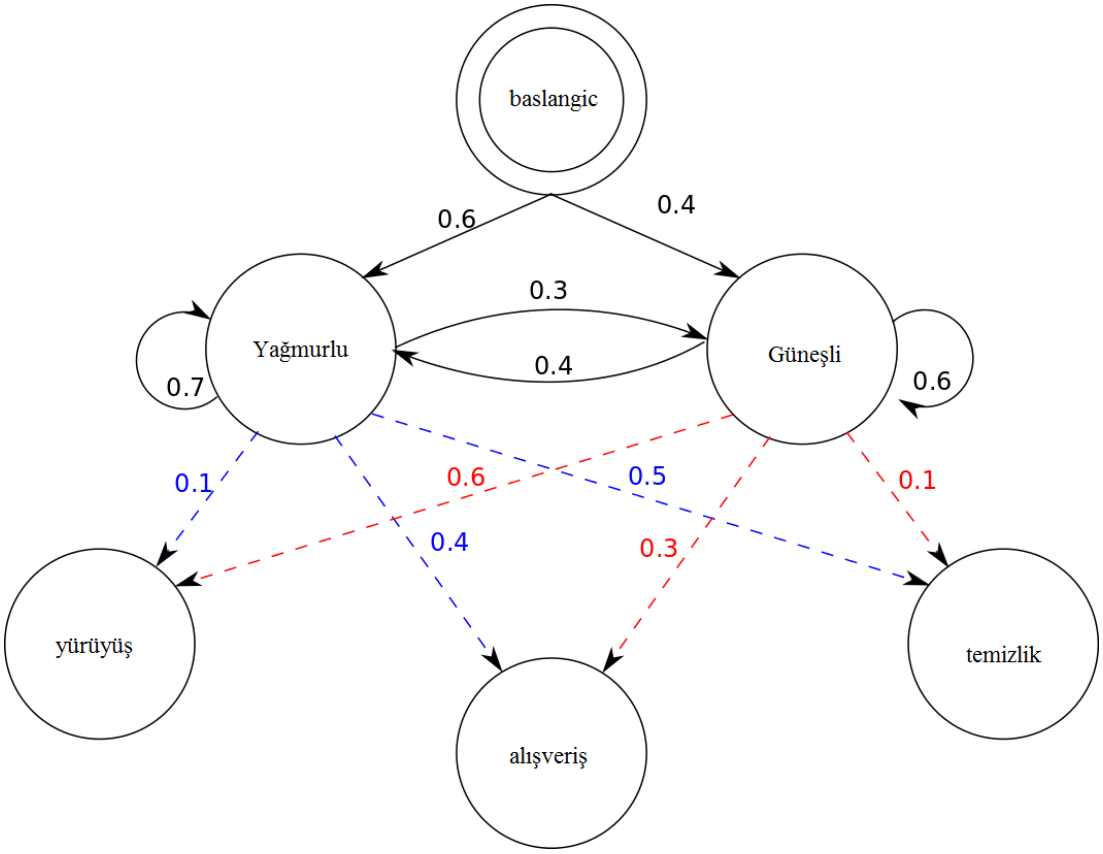


Şekil 2.1. 3 durumlu Markov zinciri [11]

Hava durumuna göre kişinin yaptığı eylemler aşağıdaki şekilde bir Markov modeliyle temsil edilebilir.

```
durumlar = ('Yağmurlu', 'Güneşli')
gozlemler = ('yürüyüş', 'alışveriş', 'temizlik')
baslangic_olasilik = {'Yağmurlu': 0.6, 'Güneşli': 0.4}
gecis_olasilik =
{
        'Yağmurlu' : {'Yağmurlu': 0.7, 'Güneşli': 0.3},
        'Güneşli' : {'Yağmurlu': 0.4, 'Güneşli': 0.6}
}
cikti_olasilik =
```

```
{
        'Yağmurlu' : {'yürüyüş': 0.1, 'alışveriş': 0.4, 'temizlik': 0.5},
        'Güneşli' : {'yürüyüş': 0.6, 'alışveriş': 0.3, 'temizlik': 0.1}
}
```



Şekil 2.2. Hava durumu-eylem Markov modeli

Bu örnekte arkaplanda çalışan durumlar(girişler) hava durumlarını, üretilen çıktılar ise gözlemlenebilir eylemleri ifade etmektedir. Eylemler serisi(çıkış dizisi) verildiğinde, eldeki SMM verilerine göre en muhtemel giriş dizisi hesaplanabilir.

SMM'lerde kullanılan olasılık hesaplamaları ilk defa Stratonovich tarafından 1950'lerin sonunda ve 1960'ta tarif edilmiştir [12]. SMM'lerin kullanıldığı ilk uygulamalardan biri KT olmuştur.

SMM'lerde yapılan hesaplamaların daha kolay anlaşılabilmesi için kafes [11] denilen bir yapı kullanılır. Fikir vermesi açısından, olasılık hesabının nasıl yapıldığı basitleştirilmiş bir

örnekle gösterilecektir. Bu örnekte, F. Jelinek'in "Statistical Methods for Speech Recognition" isimli kitabındaki [11] bilgilerden yararlanılmıştır.

SMM'lerin tanımlanması:

- $\lambda = (N, M, A, B, \Pi)$ veya $\lambda = (A, B, \Pi)$
- N: modeldeki durum sayısı
- M: farklı gözlem sembollerinin sayısı
- A: NxN'lik $A = \{a_{ij}\}$ matrisi şeklinde verilen durum geçiş olasılık dağılımı
- B: NxM'lik $B = \{b_{jk}\}$ matrisi şeklinde verilen gözlem sembol olasılık dağılımı
- Π: Başlangıç durum dağılım vektörü $= \{ \pi_i \}$

a_{ij} = i. durumdan j. duruma geçiş olasılığı

b_{jk} = j. durumun k gözlemini üretme olasılığı

Örnek SMM: $\lambda_1 = (A, B, \Pi)$

Çizelge 2.1. A geçiş matrisi

A	D0	D1	D2
D0	0,10	0,60	0,30
D1	0,40	0,20	0,40
D2	0,20	0,70	0,10

Çizelge 2.1'de, λ_1 isimli örnek Saklı Markov Modeli'nin A (geçiş) matrisi verilmiştir. Bu değerler, her satırdaki değerlerin toplamı 1 değerini verecek şekilde seçilmiştir. Örneğin basit olması için sadece 3 durumlu bir yapı seçilmiştir.

Çizelge 2.2. B çıkış matrisi

B	G0	G1	G2	G3
D0	0,54	0,12	0,21	0,13
D1	0,30	0,20	0,19	0,31
D2	0,37	0,25	0,18	0,20

Çizelge 2.2'de, λ_1 isimli örnek Saklı Markov Modeli'nin B (çıkış) matrisi verilmiştir. Bu değerler, her satırdaki değerlerin toplamı 1 değerini verecek şekilde seçilmiştir.

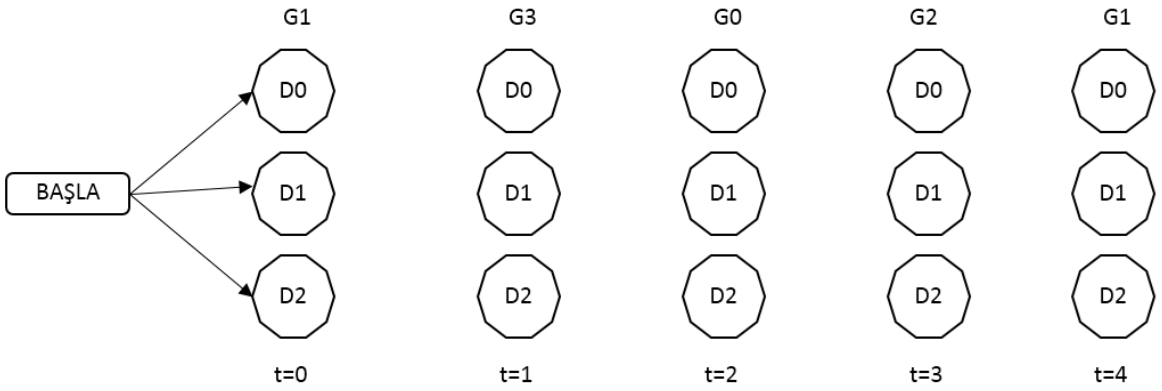
Çizelge 2.3. Π başlangıç vektörü

π	i
D0	0,06
D1	0,73
D2	0,21

Çizelge 2.3'te, λ_1 isimli örnek Saklı Markov Modeli'nin Π (başlangıç) vektörü verilmiştir. Π vektörü, t=0 anında, hangi duruma geçiş yapılma olasılığının ne olduğunu gösterir. Bu değerler, toplamları 1 değerini verecek şekilde seçilmiştir.

Bu SMM için O = "G1 G3 G0 G2 G1" gözlem dizisi verilsin. Modelin O gözlem dizisini üretme olasılığı ve bu gözlem dizisini üretmesi en muhtemel olan durum dizisinin nasıl hesaplandığı gösterilecektir.

1. adım (Şekil 2.3):



Şekil 2.3. Olasılık hesabı (1)

Şekil 2.3'te görüleceği gibi, ilk adımda hangi duruma geçiş yapılacağına karar vermek gerekir. Bu adımda üretilecek olan çıkış G1'dir. Π matrisindeki D0 durumuna geçme olasılığı (0,06) ile, B matrisindeki D0 durumunun G1'i üretme olasılığının çarpımı; t=0 anında D0 durumuna geçiş yapılma olasılığını (P0) verir. Benzer şekilde diğer geçiş olasılıkları da (P1, P2…) hesaplanır. En yüksek olasılık hangisi ise o duruma geçiş yapılır.

P0 = PI[0] x B[0 , 1] = 0,06 x 0,12 = 0,0072

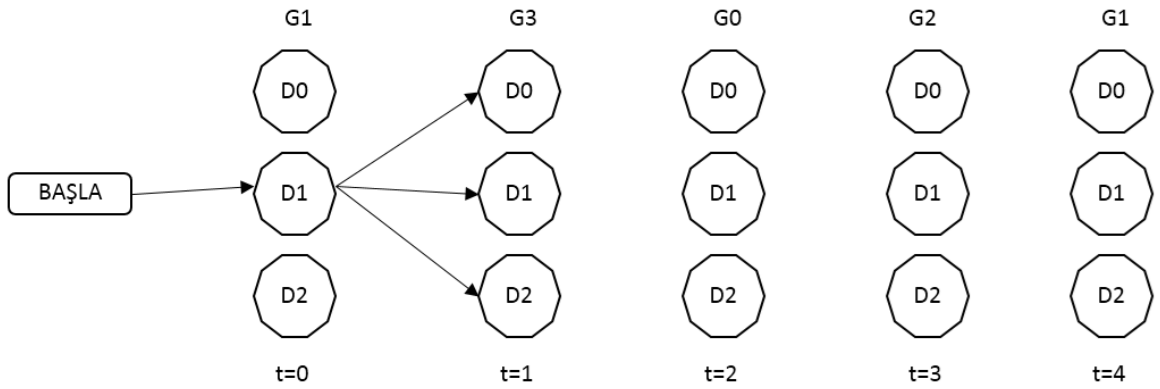
*P1 = PI[1] x B[1 , 1] = 0,73 x 0,20 = 0,1460

P2 = PI[2] x B[2 , 1] = 0,21 x 0,25 = 0,0525

Hafıza değişkeni mem, başlangıçta 1 kabul edilir (mem=1). Bundan sonraki her adımda, P_{ij} olasılığını maksimize eden P değeri ile çarpılır.

mem=1
mem = mem x P1 = 1 x 0,1460 = 0,1460

2. adım (Şekil 2.4):

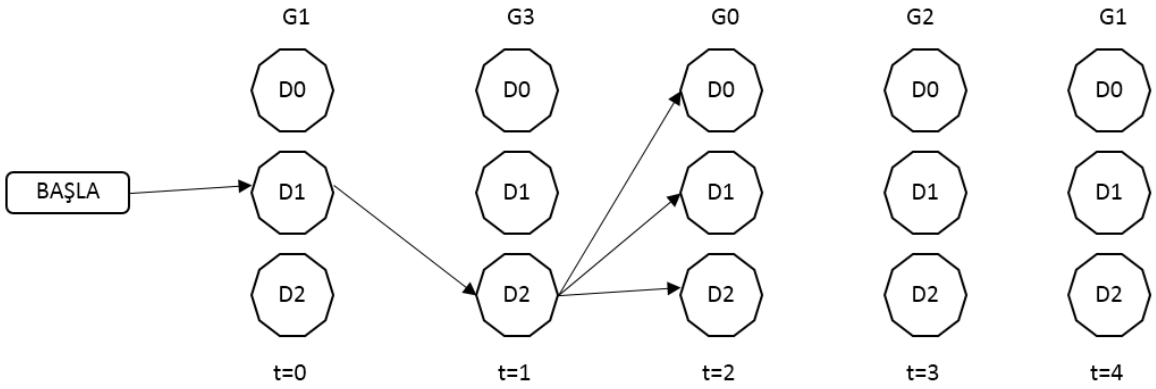


Şekil 2.4. Olasılık hesabı (2)

P0 = A[1 , 0] x B[0 , 3] = 0,40 x 0,13 = 0,052
P1 = A[1 , 1] x B[1 , 3] = 0,20 x 0,31 = 0,062
*P2 = A[1 , 2] x B[2 , 3] = 0,40 x 0,20 = 0,080

mem = 0,1460
mem = mem x P2 = 0,1460 x 0,080 = 0,01168

3. adım (Şekil 2.5):



Şekil 2.5. Olasılık hesabı (3)

P0 = A[2 , 0] x B[0 , 0] = 0,20 x 0,54 = 0,108

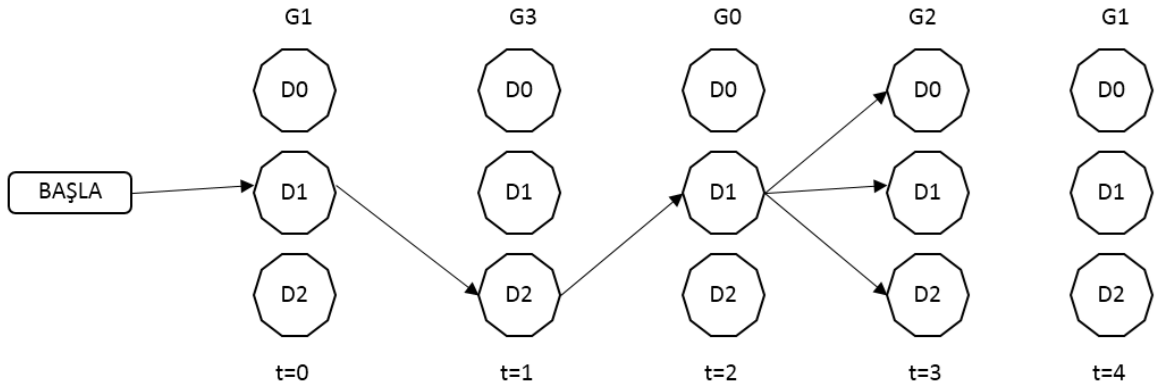
*P1 = A[2 , 1] x B[1 , 0] = 0,70 x 0,30 = 0,210

P2 = A[2 , 2] x B[2 , 0] = 0,10 x 0,37 = 0,037

mem = 0,01168

mem = mem x P1 = 0,01168 x 0,210 = 0,0024528

4. adım (Şekil 2.6):



Şekil 2.6. Olasılık hesabı (4)

*P0 = A[1 , 0] x B[0 , 2] = 0,40 x 0,21 = 0,084

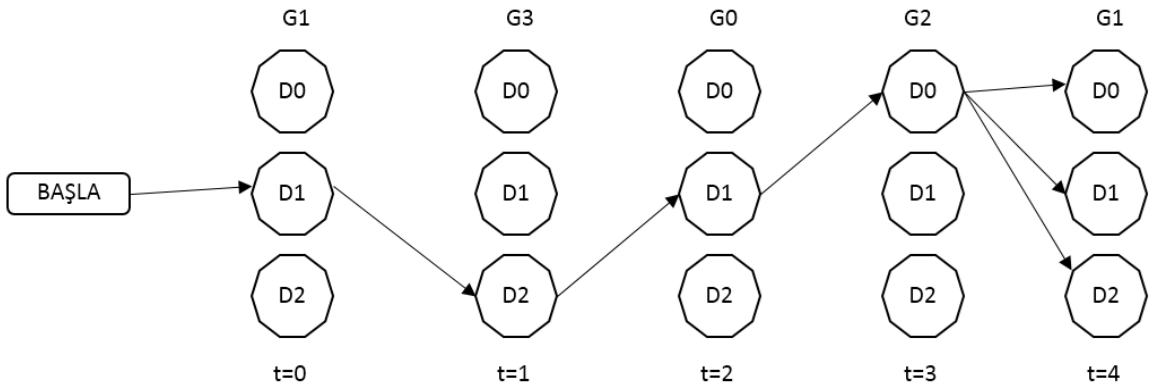
P1 = A[1 , 1] x B[1 , 2] = 0,20 x 0,19 = 0,038

P2 = A[1 , 2] x B[2 , 2] = 0,40 x 0,18 = 0,072

mem = 0,0024528

mem = mem x P0 = 0,0024528 x 0,084 = 0,0002060352

5. adım (Şekil 2.7):



Şekil 2.7. Olasılık hesabı (5)

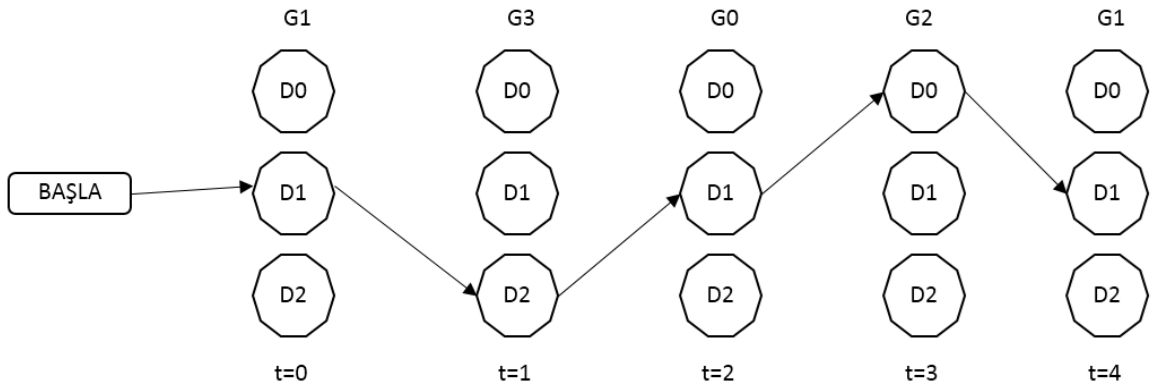
P0 = A[0 , 0] x B[0 , 1] = 0,10 x 0,12 = 0,012

*P1 = A[0 , 1] x B[1 , 1] = 0,60 x 0,20 = 0,120

P2 = A[0 , 2] x B[2 , 1] = 0,30 x 0,25 = 0,075

mem = 0,0002060352

mem = mem x P01 = 0,0002060352 x 0,120 = 0,000024724224



Şekil 2.8. Olasılık hesabı (6)

“G1 G3 G0 G2 G1” gözlem dizisini üretmesi en muhtemel durum dizisi (Şekil 2.8):

“D1 D2 D1 D0 D1”

Bu dizinin “merhaba” gözlem dizisini üretme olasılığı:

0,000024724224

3. KONUŞMA VERİTABANI OLUŞTURMA SÜRECİ

Bu bölümde, Türkçe dili için bir konuşma veritabanı oluşturma aşamaları adım adım incelenecektir. Öncelikle dünya çapında en çok kullanılan konuşma veritabanlarından biri olan TIMIT hakkında bilgi verilecektir.

TIMIT, Defense Advanced Research Projects Agency (DARPA) tarafından yaptırılan, başta Texas Instruments (TI) ve Massachusetts Institute of Technology (MIT) olmak üzere pek çok kuruluş tarafından geliştirilen, dünya çapında çok sık kullanılan [13] bir konuşma veritabanıdır.

Amerikan İngilizce'sinin başlıca 8 lehçesini konuşan 630 kişiye 10'ar cümle okutulmasıyla oluşturulmuştur [14].

Aşağıda TIMIT veritabanından örnekler verilmiştir.

Cümleler:
0 46797 She had your dark suit in greasy wash water all year.

Kelimeler:
3050 5723 she
5723 10337 had
9190 11517 your
11517 16334 dark
16334 21199 suit
21199 22560 in
22560 28064 greasy
28064 33360 wash
33754 37556 water
37556 40313 all
40313 44586 year

Fonemler:

0 3050 h#
3050 4559 sh
4559 5723 ix
5723 6642 hv
6642 8772 eh
8772 9190 dcl
9190 10337 jh
10337 11517 ih
11517 12500 dcl
12500 12640 d
12640 14714 ah
14714 15870 kcl
15870 16334 k
16334 18088 s
18088 20417 ux
20417 21199 q
21199 22560 en
22560 22920 gcl
22920 23271 g
23271 24229 r
24229 25566 ix
25566 27156 s
27156 28064 ix
28064 29660 w
29660 31719 ao
31719 33360 sh
33360 33754 epi
33754 34715 w
34715 36080 ao
36080 36326 dx
36326 37556 axr
37556 39561 ao
39561 40313 l

40313 42059 y
42059 43479 ih
43479 44586 axr
44586 46720 h#

İlk iki sütunda ifadenin söylendiği zaman aralığının başlangıç ve bitiş noktaları görülmektedir. Üçüncü sütundan itibaren ise konuşulan ifade yer almaktadır. Fonem düzeyindeki damgalamaya özellikle sözcük altı çalışmalarda ihtiyaç duyulmaktadır.

Veritabanı oluşturulurken seçilecek fonem seti çok önemlidir. Fonem seti o dildeki seslerden herhangi birini boşta bırakmayacak kadar geniş, performansı düşürmeyecek kadar da küçük olmalıdır. Şekil 3.1'de, ODTÜ Türkçe Mikrofon Konuşması Veritabanı'nda kullanılan METUBET isimli fonem setindeki fonemleri IPA'daki fonemlerle karşılaştırılması verilmiştir.

Çizelge 3.1. METUBET-IPA fonem tablosu [5]

IPA	METUbet	Example
ɑ	AA	*a*nı
a	A	l*a*f
e	E	*e*lma
ɛ	EE	der*e*
i	IYG	*i*ğde
IY	IY	s*i*m*i*t
Ï	I	*ı*s*ı*
ɔ	O	s*o*ru
o	OG	*o*ğlak
ʊ	U	k*u*lak
u	UG	*u*ğ*u*r
Œ	OE	örtü
ɸ	OEG	öğren
y	UE	ümit
y	UEG	düğme
b	B	*b*al
d	D	*d*e*d*e
l	L	*l*eylek
ɬ	LL	ku*l*
m	M	da*m*
n	NN	a*n*
ŋ	N	sü*n*gü
p	P	i*p*
r	R	*r*af
ɾ	RR	ı*r*mak
ɣ	RH	bi*r*
s	S	*s*e*s*
ʃ	SH	aşı
t	T	ü*t*ü
v	VV	*v*ar
ʋ	V	ta*v*uk
j	Y	*y*at
z	Z	a*z*ık
ʐ	ZH	yo*z*
ʤ	C	*c*am
ʧ	CH	seçim
f	F	*f*asıl
ʒ	J	müj*d*e

METUBET seti Türkçe dilindeki tüm fonemleri içermektedir. Hatta çoğu konuşmacı tarafından farkı bilinmeyen, sıklıkla birbirinin yerine kullanılan fonemler bile ayrıca tanımlanmıştır. Örneğin *e*lma kelimesindeki e harfi ile, dere kelimesindeki e harfi için farklı fonemler tanımlanmıştır. Bu, elbette dildeki fonem çeşitliliğini eksiksiz olarak temsil

edebilmek için gereklidir. Ancak, KT görevi için ekstra hesaplama maliyetlerine neden olmaktadır.

İkinci bölümde verilen SMM olasılık hesaplama örneğinde farklı çıkış sembolü sayısı M=4, arkaplanda çalışan durum sayısı N=3 idi. KT probleminde, her bir üçlüses için ayrı bir SMM tanımlanmaktadır. Üçlüseslerin sayısının artması da arama uzayı'nın (search space) genişlemesine neden olur. Fonem sayısı arttıkça sistemin hesaplama yükü artacaktır.

Veritabanı oluşturulurken mümkün olduğunca az sayıda fonem tanımlanmasında performans açısından fayda vardır. Türkçe'nin *genelde* yazıldığı gibi okunan bir dil olmasından da yola çıkarak, fonem seti alfabe ile birebir alınacaktır.

Fonem setini alfabe ile birebir almamızın nedenlerinden biri de, Türkçe dili için objektif telaffuz kurallarına göre oluşturulmuş bir telaffuz sözlüğü bulunmamasıdır. ODTÜ Türkçe Mikrofon Konuşması Veritabanı'nında kullanılan telaffuz kuralları, Ergenç'in Konuşma Dili ve Türkçe'nin Söyleyiş Sözlüğü [15] isimli kitabından alınmıştır. Bu kitapta belirtilen bazı telaffuz kuralları şu şekildedir:

- Kahve ~ /ka:ve/
- Ahmet ~ /a:met/
- Bir dakika ~ /bi dakka/
- Geliyor mu ~ /geliyo mu/
- Söyledim ~ /sö:ledim/
- Öyle ~ /ö:le/
- Onbaşı ~ /ombaşı/
- Yanlış ~ /yannış/
- Dinlemek ~ /dinnemek/
- Yatsı ~ /yassı/
- Ağlayacak ~ /ağlıyacak/
- Kollayan ~ /kollıyan/
- Güçlük ~ /güşlük/
- Kapayacak ~ /kapi:cak/
- Olacak ~ /olucak/

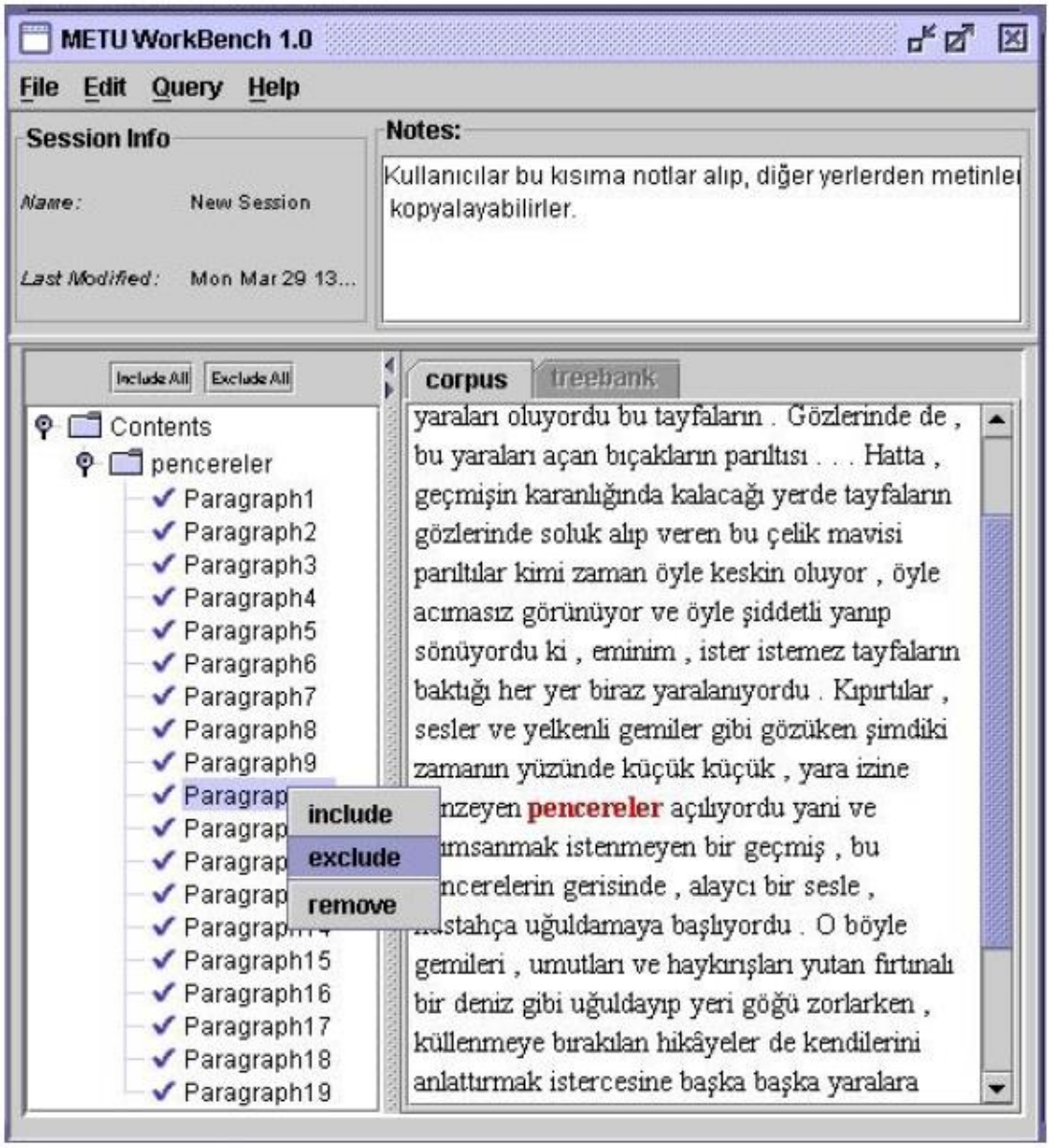
- Vermeyecek ~ /vermi:cek/

Yapılan tez çalışmasında okuma konuşması merkeze alındığı için, bu kurallara göre hareket etmek doğru olmazdı. Çünkü bunlar, konuşma dilinde bile nadiren görülen telaffuzlardır.

Veritabanı oluşturulurken seslendirilecek cümlelerin seçimi de KT performansını etkileyen bir faktördür. Seçilen cümlelerde geçen üçlüsesler, o dilde var olan, en çok kullanılan üçlüsesleri iyi bir şekilde temsil etmelidir.

Kullanılacak cümlelerin rastgele seçilmemesi gerekir. Bu nedenle, tez çalışması kapsamında, Türkçe dilinde en çok kullanılan üçlüsesleri tespit edebilmek amacıyla 2 milyon kelimeye yaklaşan (1993680) ODTÜ Türkçe Derlemi (OTD) [16] üzerinde çeşitli hesaplamalar yapılmıştır.

Derlemle beraber dağıtılan workbench yazılımında (Resim 3.1) kelime seviyesinde hesaplamalar yapılabilmektedir. Ancak fonem düzeyinde, kelime altı hesaplamalara destek vermemektedir. Bu nedenle, bu tez çalışması için yazılan programlar ile Türkçe'nin kelime altı istatistik hesaplamaları yapılmıştır.



Resim 3.1. OTD workbench yazılımı [17]

Derlem geliştirme süreci ile ilgili Say [16] şunları söylemektedir:

“Derlem yapımı, metin toplama, işaretleme ve sorgu yazılımının geliştirilmesi olmak üzere üç ana süreci kapsamaktadır.

Metin toplama süreci, derleme alınacak eserlerin telif hakkı sahiplerinden yazılı izin alınması ile başlayıp, metinlerin bir sonraki süreç olan işaretleme süreci öncesi elektronik ortamda "unicode salt metin" formatında hazır bulunmasıyla sona erer. İşaretleme süreci ODTÜ Türkçe Derlemi için, bilgisayarda geliştirilen özel yazılımlarla yazım özelliklerinin

standartlara uygun olarak işaretlenmesini içerir; ODTÜ-Sabancı Ağaç Yapılı altderlemi içinse bu süreçte öncelikle proje kapsamı dışında geliştirilmiş bir yazılımla tüm metinlerin biçimbirimsel çözümlemelerinin yapılması, daha sonra yine proje kapsamında geliştirilen bir yazılımla biçimbirimsel çözümlemelerin gerekiyorsa düzeltilmesi ve indirgenmesi (İng. disambiguation) ve sözdizimsel çözümlemenin yapılması yeralmaktadır. Ayrıca hem eşdizimli öbeklerin (İng. collocations) hem de sözdizimsel öğelerin otomatik işaretlenmesine yönelik bir takım yazılımlar da bu süreç dahilinde gerçekleştirilmiştir. ODTÜ Türkçe Derlemi kapsamında ayrıca derlem üzerinde sözcük ve öbek bazında kolay sorgu yapılabilmesini sağlayacak bir yazılım gerçekleştirilmiştir. Tüm bu süreçler, kendi içinde planlama (sürece göre standart geliştirme, yazılım tasarımı, eğitim gibi değişik görevleri içermektedir), gerçekleştirim ve kontrol altsüreçlerini gerektirmektedirler.”

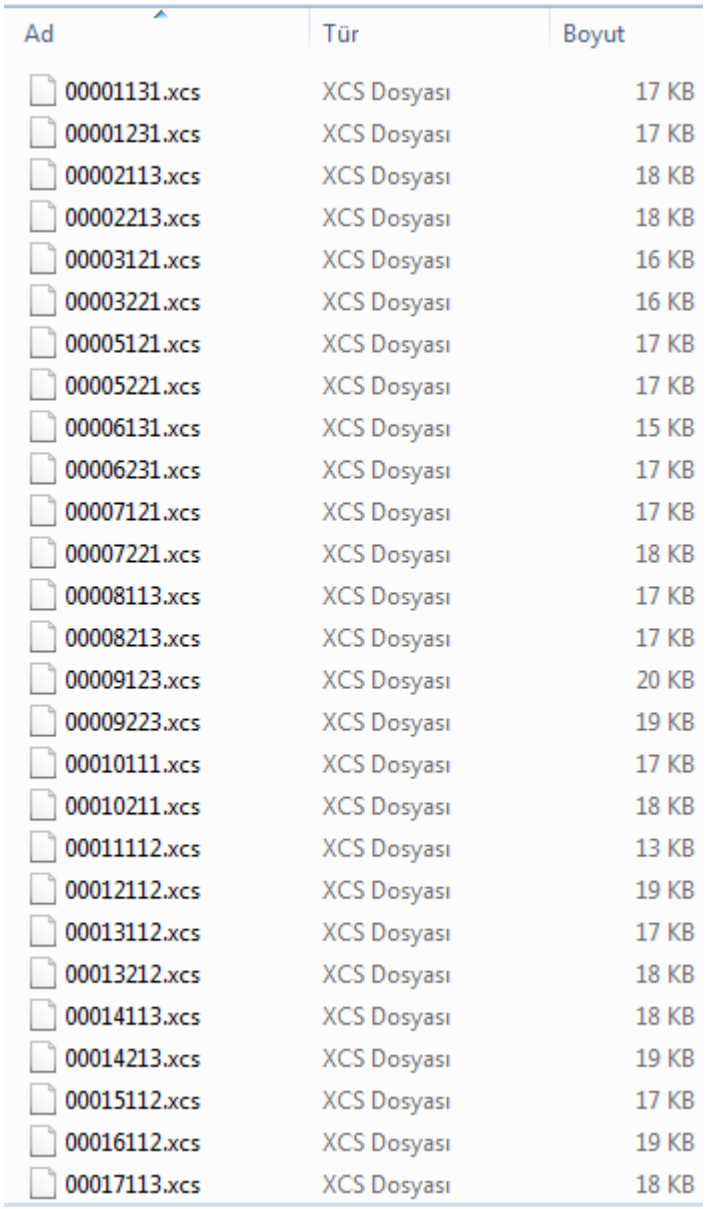
ODTÜ Türkçe Derlemi’nin toplanması aşamasında hangi kaynakların hangi oranda kullanıldığı Çizelge 3.2’de verilmiştir.

Çizelge 3.2. OTD kaynak dağılımı [16]

Kaynak	**%**
Köşe Yazısı	8%
Haber	42%
Roman	13%
Öykü	11%
Makale	8%
Deneme	7%
Araştırma	5%
Gezi	2%
Söyleşi	1%
Diğer	3%
Toplam	100%

Bu noktadan sonra OTD’nin, tez çalışmasının amacı doğrultusunda düzenlenmesi için yapılanlar anlatılacaktır.

Resim 3.2’de görüldüğü gibi, OTD’nin içerisindeki metinler, 998 dosyaya bölünmüş halde bulunmaktadır.



Ad	Tür	Boyut
00001131.xcs	XCS Dosyası	17 KB
00001231.xcs	XCS Dosyası	17 KB
00002113.xcs	XCS Dosyası	18 KB
00002213.xcs	XCS Dosyası	18 KB
00003121.xcs	XCS Dosyası	16 KB
00003221.xcs	XCS Dosyası	16 KB
00005121.xcs	XCS Dosyası	17 KB
00005221.xcs	XCS Dosyası	17 KB
00006131.xcs	XCS Dosyası	15 KB
00006231.xcs	XCS Dosyası	17 KB
00007121.xcs	XCS Dosyası	17 KB
00007221.xcs	XCS Dosyası	18 KB
00008113.xcs	XCS Dosyası	17 KB
00008213.xcs	XCS Dosyası	17 KB
00009123.xcs	XCS Dosyası	20 KB
00009223.xcs	XCS Dosyası	19 KB
00010111.xcs	XCS Dosyası	17 KB
00010211.xcs	XCS Dosyası	18 KB
00011112.xcs	XCS Dosyası	13 KB
00012112.xcs	XCS Dosyası	19 KB
00013112.xcs	XCS Dosyası	17 KB
00013212.xcs	XCS Dosyası	18 KB
00014113.xcs	XCS Dosyası	18 KB
00014213.xcs	XCS Dosyası	19 KB
00015112.xcs	XCS Dosyası	17 KB
00016112.xcs	XCS Dosyası	19 KB
00017113.xcs	XCS Dosyası	18 KB

Resim 3.2. OTD dosya yapısı

Xcs uzantılı dosyalar bir metin editörüyle açıldığında Resim 3.3’deki gibi bir görüntü ortaya çıkmaktadır.

00223166.xcs

```
<cesDoc>
<cesHeader>
<fileDesc>
<titleStmt>
<h.title>00223160</h.title>
</titleStmt>
<extent>
<wordcount>1428</wordcount>
<bytecount>12049</bytecount>
</extent>
<sourceDesc>
<biblStruct>
<analytic>
<h.title>Balık her şeyi bilir</h.title>
<h.author>Zafer Kızılkaya</h.author>
</analytic>
<monogr>
<h.title></h.title>
<h.author></h.author>
<edition></edition>
<imprint>
<publisher>Atlas</publisher>
<pubdate>Ağustos 2001</pubdate>
<pubplace>İstanbul</pubplace>
</imprint>
<idno>1300-5901</idno>
</monogr>
</biblStruct>
</sourceDesc>
</fileDesc>
<profileDesc>
<textClass>
<catRef>Gezi Yazısı</catRef>
</textClass>
</profileDesc>
<revisionDesc>
<change>
<changeDate>15.11.2001</changeDate>
<respName>Deniz</respName>
<h.item>Yazım yanlışı olan kelimeler düzeltildi.</h.item>
</change>
</revisionDesc>
</cesHeader>
<text>
<body>
<p><hi>Yüz</hi> bin yıl önce Afrika'dan ayrılan 100 kadar insan, bugün Afrika dışın
<p>Altın ve elmasın getirdiği muazzam gelirin ötesinde Afrika'da daha neler olduğun
<p>Aracımız kumsalda kırılan dalgaların son nefesi üzerinde ilerlerken, yüzlerce, k
<p><hi>Bir</hi> balıkçıya sorduğumuz sorulara ilginç yanıtlar alıyoruz. Eskiden kıy
<p>Kıyıdaki binlerce pirogu görünce bir avda ne kadar balık yakaladıklarını soruyor
<p>Böylece yasal avlanma izni olan yabancıların açık denizde yaptığı aşırı avcılık,
<p>Kıyı boyunca teknelerin arasında yürümeye devam ediyoruz. Küçük teknelerle yapıl
<p>Dakar Konsolosumuz Özgür Çınar'ı aradığımızda, bize birkaç büyük balıkçılık firm
<p>Yabancı gemiler, soğuk hava depolarını da beraberlerinde taşıdığından, yakaladık
<p>Son olarak gittiğimiz Çin-Senegal ortaklı büyük bir firmanın Sengalli yöneticisi
</body>
</text>
</cesDoc>
```

Resim 3.3. Xcs dosyalarının içeriği

Bu dosyalar incelendiğinde, <body>…</body> etiketlerinin arasında kalan kısmın metin olduğu görülmektedir. Bu nedenle, önce bu etiketlerin arasındaki kısımlar çekilmiş, daha sonra 998 dosyadan elde edilen bu veriler birleştirilmiştir.

Etiketleme işlemleri, birçok farklı amaç için oldukça önemlidir. Ancak bizim amaca uygun olmadığından, verileri işlemeden önce bunların temizlenmesi gerekmektedir. Metin içerisinde bulunan <corr>, <p>, <ptr>, <lg>, <note>, <item>, <q>, <hi>… gibi etiketler C# dilinde yazılan kodlar vasıtasıyla temizlenmiştir.

Konuşma veritabanı için fonem bazında istatistiki çalışmaların yapılabilmesi için, metin içerisindeki "nokta ve virgül dışındaki" tüm özel karakterlerin temizlenmesi gerekir. Parantez, tırnak, çift tırnak, noktalı virgül, iki nokta, üç nokta gibi karakterlerin tümü temizlenmiştir. Parantezler kaldırılırken parantezlerin arasında kalan metinler de çıkarılmıştır. Noktalı virgüller virgüle dönüştürüldü. Gereken yerlerde, metni mümkün olduğunca koruyarak kararlar verilmiştir.

Sayıların da harflere dönüştürülmesi gerekmektedir. Çünkü yapılacak çalışma çerçevesinde sayısal ifadelerden ziyade, bunların okunuşları anlam ifade etmektedir. Sayısal ifadeler farklı şekilde okunabilmektedir. Kimi tarih, kimi para, kimi sayı ifade edebilmektedir. Yapılan dönüştürme işlemi şu şekildedir:

- 16.07.1997 -> on altı temmuz bin dokuz yüz doksan yedi
- 25000 -> yirmi beş bin
- 12.500.000 -> on iki milyon beş yüz bin

Tüm düzenlemeler yapıldıktan sonra elde edilen dosyanın görüntüsü Resim 3.4'te verilmiştir. Her satıra bir cümle gelecek şekilde düzenleme yapılmıştır. Bu işlemin sonucunda 170159 cümle elde edilmiştir.

```
koca bir duvar taşıyordun yüreğinde kimsenin aşamayacağı aşmaya ce
dışa karşı güçlüydü ama içe kendi yüreğine yıkılmak üzereydi.
anılarla örülmüş acılarla harçlanmış bu duvara tırmanmak onu aşabi
vazgeçmek kolaydı ertelemek de.
ama tırmanmaya başlandı mı bitirilmeli.
çünkü her seferinde acımasız bir geriye dönüş vardı.
bıraktığın her sefer bir başlangıca gebeydi.
bir aşsaydım bu duvarı benim olacaktın kucaklayacaktın beni.
kırgınlıkların korkuların eriyip gidecekti hepsi benim olacak bana
ben kıvranacaktım ben acı çekip işkencelere gönüllü katlanacaktım
benim yüreğim de duvar taşıyordu.
aşmaya yeltenen olmadı.
ben bu duvarı taşıyan birçok insan gördüm ve aşmaya değil yıkmaya
sen de haberdar değildin ve ben hayatımda ilk kez yıkmaya değil aş
izin vermiyor engeller koyuyordun.
dikenli tellerle çeviriyordun bu duvarı.
yaralanıyordum tırmanırken kanıyordum.
kırılıyordum acıyordum ama bırakmıyordum.
korkmuyordum etimin kesilmesinden duygularımın can çekişmesinden.
yalnızca tırmanıyordum ardıma bakmadan belki de bakmaya cesaret ed
sandalyemin tekerleklerini çevirerek koltuğunun önüne gelmiştim.
gülümsüyordun.
ellerimi bacaklarına koyduğumda sanki yeniden büyük bir ağacın en
sanki lunaparklarda insanın içini ürperten dev oyuncaklardan biris
ışıklar yanıp sönüyor yüksek tonda sevmediğim bir müzik çalıyordu.
çığlıklar korku mutluluk bozma bu anı konuşma gülümseme hiçbir şey
konuşmadın kızmadın hiçbir şey yapmadın.
bir terslik vardı.
kaçmaktı en iyisi.
evet kaçmak istiyordum.
en çabuk nasıl nasıl gidebilirdim.
nasıl hızla uzaklaşabilirdim.
koşsam gücüm yeter miydi.
```

Resim 3.4. Düzenleme sonrası elde edilen dosya

KT probleminde her fonem kendinden önceki fonemden etkilenmekte ve kendinden sonraki fonemi etkilemektedir. Bu nedenle, üçlüses denilen bir yapı kullanılmaktadır.

Örnek cümle:

“Işıklar yanıp sönüyor, yüksek tonda sevmediğim bir müzik çalıyordu.”

Cümlenin üçlüses karşılığı Çizelge 3.3’te verilmiştir.

Çizelge 3.3. Örnek cümlenin üçlüses karşılığı

Sıra	Üçlüses	Sıra	Üçlüses	Sıra	Üçlüses	Sıra	Üçlüses
1	SIL-ı+ş	16	n-ü+y	31	d-a+s	46	m-ü+z
2	ı-ş+ı	17	ü-y+o	32	a-s+e	47	ü-z+i
3	ş-ı+k	18	y-o+r	33	s-e+v	48	z-i+k
4	ı-k+l	19	o-r+SP	34	e-v+m	49	i-k+ç
5	k-l+a	20	r-SP+y	35	v-m+e	50	k-ç+a
6	l-a+r	21	SP-y+ü	36	m-e+d	51	ç-a+l
7	a-r+y	22	y-ü+k	37	e-d+i	52	a-l+ı
8	r-y+a	23	ü-k+s	38	d-i+ğ	53	l-ı+y
9	y-a+n	24	k-s+e	39	i-ğ+i	54	ı-y+o
10	a-n+ı	25	s-e+k	40	ğ-i+m	55	y-o+r
11	n-ı+p	26	e-k+t	41	i-m+b	56	o-r+d
12	ı-p+s	27	k-t+o	42	m-b+i	57	r-d+u
13	p-s+ö	28	t-o+n	43	b-i+r	58	d-u+SIL
14	s-ö+n	29	o-n+d	44	i-r+m		
15	ö-n+ü	30	n-d+a	45	r-m+ü		

SIL fonemi ses dosyasının başında ve sonunda bulunan sessizliğe, SP ise *genellikle* virgül nedeniyle oluşan kısa duraklamaya karşılık gelmektedir.

Bir kelimeyi üçlüseslere dönüştüren örnek C# kodları aşağıdaki gibidir:

```
for (int i = 0; i <= kelime.Length - 3; i++)
{
        ilk = kelime.Substring(i, 1);
        orta = kelime.Substring(i + 1, 1);
        son = kelime.Substring(i + 2, 1);
        sonuc += ilk + "-" + orta + "+" + son + "\n";
}
```

Bu işlem sonucunda 12.5 milyon civarında üçlüses elde edilmiştir. Bunların içinde 15060 eşsiz üçlüses olduğu hesaplanmıştır. 29 harf + SIL + SP = 31 fonemden teorik olarak en fazla 31*31*31=29791 üçlüses elde edilebilir. Elde edilen 2 milyon kelimelik metin, bunların sadece %44'ünün Türkçe'de kullanıldığını göstermektedir.

Daha sonra bu eşsiz üçlüseslerin OTD içinde geçme sayıları (frekans) hesaplanmıştır. Elde edilen verilerin bir kısmı Çizelge 3.4'te verilmiştir.

Çizelge 3.4. Üçlüses istatistikleri

Sıra	Üçlüses	Frekans	%	Sıra	Üçlüses	Frekans	%
1	l-a+r	101275	0,81281179	21	d-e+n	33649	0,27005978
2	l-e+r	81497	0,65407773	22	a-d+a	33376	0,26786874
3	e-r+i	72860	0,58475899	23	r-ı+n	32699	0,26243528
4	b-i+r	71899	0,57704621	24	e-d+i	31854	0,25565349
5	a-r+ı	64277	0,51587365	25	e-n+i	31154	0,25003543
6	a-r+a	54161	0,43468476	26	e-d+e	30713	0,24649606
7	n-d+a	50896	0,40848056	27	e-s+i	30639	0,24590215
8	i-n+i	47937	0,38473225	28	n-l+a	28261	0,22681683
9	y-o+r	44454	0,35677843	29	n-i+n	28150	0,22592596
10	n-d+e	43322	0,34769324	30	n-ı+n	27681	0,22216187
11	ı-n+ı	39067	0,31354350	31	i-y+e	26907	0,21594991
12	r-i+n	38685	0,31047765	32	i-n+e	26903	0,21591780
13	a-s+ı	38014	0,30509235	33	a-l+a	26586	0,21337363
14	i-l+e	37664	0,30228332	34	e-l+e	26366	0,21160796
15	ı-n+d	37086	0,29764441	35	d-a+n	26323	0,21126285
16	i-n+d	36758	0,29501196	36	i-l+i	25973	0,20845382
17	a-y+a	36641	0,29407294	37	s-ı+n	25090	0,20136705
18	a-n+ı	35393	0,28405675	38	e-r+e	25082	0,20130284
19	a-m+a	33714	0,27058145	39	o-l+a	25013	0,20074906
20	l-a+n	33706	0,27051725	40	a-n+a	24433	0,19609411

Bu veriler incelendiğinde –ler, -lar gibi eklerin, en çok kullanılan üçlüsesler olduğu görülmektedir. Bu, Türkçe dilinin eklemeli yapısının doğal bir sonucudur.

Son işlem olarak, elde edilen bu bilgiler ışığında, OTD içindeki 170160 cümle arasında en sık kullanılan fonemleri içeren cümleler bulundu. Her cümleye bir skor verildi (Çizelge 3.5). Konuşmacılar uzun cümleleri telaffuz etmekte sıkıntı çektiği için, 60 karakterden uzun olan cümleler elendi.

Çizelge 3.5. En yüksek skor alan 35 cümle

Sıra	Skor	Cümle
1	1103813,50	saray ileri gelenlerinin gözde konuklarından birisiydi
2	1043688,50	ayrıca kendilerinden olmayanları kafir ilan ediyorlar
3	1040601,33	başkalarından alacakları önerileri de sınamalılar
4	1040304,00	onların birbirlerini görmelerini sağlamaya çalışacaktım
5	1039764,50	yaz gecelerinin birinde onların seslerini duyacaksınız
6	1035147,00	birileri demir kapılar ardında yarınları düşünüyor
7	1032597,00	rika çoraplarının kolilerini taşıdığı günlerden birindeydi
8	1020624,50	bekliyorum ilaçların derinlere ilerlemesini genzime kadar
9	1014137,00	hıçkırıklarımın arasında söylediklerini dinlemeye başladım
10	1013928,50	sonra yeni kiracılara ya da başka birilerine kedilik ederim
11	1012334,50	bir yandan da silah denetçilerinin çalışmaları devam ediyor
12	1008603,50	ancak taklitler kendilerini kutularından ele veriyor
13	1001613,00	sekreterimden bana kitaplarından birini almasını istedim
14	1001283,50	ikizlerin birbirlerine çok bağlı olduklarını biliyordum
15	999605,50	bankalarını hortumlamayanların şirketleri de batabiliyor
16	991311,83	öteki masalardakiler ise balıklarını daha yeni yiyorlardı
17	990389,50	pencerelerini topladığında yedinin katlarına ulaşıyordu
18	987480,00	satır aralarında neler söylenmek istendiğini anlıyorum
19	975870,00	cenin pozunda bacaklarının altında kilitledi ellerini
20	975785,50	osmanlıların kendilerini türk saymadıkları biliniyor
21	974664,00	bunların da taşın altına ellerini sokmaları gerekiyor
22	970853,00	yerlerini yeniden değiştirdiğimde ağlamaları hemen kesildi
23	966909,00	bunların en önemlilerinden birisi de korkunun aşılmasıdır
24	966899,00	o şiirlere kendi hayatlarında bir karşılık bulamamaları
25	963690,50	okulların mesleklerin ya da derneklerin hangi birinden
26	956862,00	ancak bu kadınların genetik özelliklerinden kaynaklanıyor
27	955902,58	bir zamanlar kelaynakların düşmanlarından biri avcılardı
28	951734,00	ama bu bu yaptıkları yanlışların en önemsizlerinden biri
29	951486,00	yan yana yürürlerken kollarının birbirine değişinden
30	950631,67	artık birbirlerini bir aile olarak görmeye başlarlar
31	948992,50	yapılanlar yapılabileceklerin yanında devede kulak kalır
32	947687,50	kağıtların arasında anneannelerinin bir notunu buldu
33	947236,00	birden çocukların defterlerle kalemleri hızlanan kapışması
34	944468,00	pazar günü akranlarının uzundereye indiklerini öğreniyor
35	943130,00	o gün kömür madenlerinin karanlıklarında canı çıkmıştı

Cümlelere skor vermek için kullanılan C# kodları aşağıdaki gibidir:

```
double skor;
int gecme_sayisi;
```

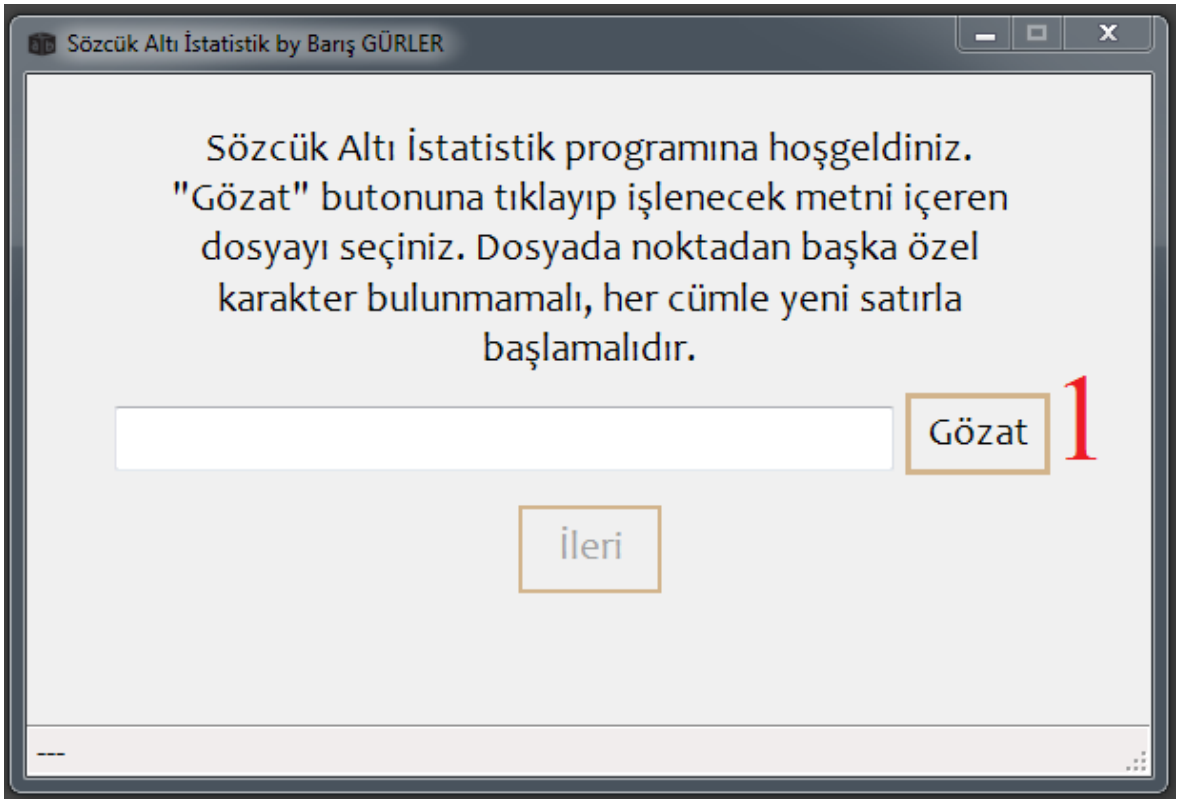
```
string[] uclusesler;
progressBar1.Maximum = cumleler.Length;
for (int i = 0; i < cumleler.Length; i++)
{
        skor = 0;
        uclusesler = ortak.ucluses_uret(cumleler[i].Replace(" ", String.Empty));
        for (int j = 0; j < uclusesler.Length; j++)
        {
                gecme_sayisi = 1;
                for (int k = 0; k < j; k++)
                {
                        if (uclusesler[j].Equals(uclusesler[k])) gecme_sayisi++;
                }
                skor += ortak.ucluses_frekans[uclusesler[j]] / (double)gecme_sayisi;
        }
        ortak.skorlar[i] = skor;
        if (i % 1000 == 0)
        {
                progressBar1.Value = i;
                Application.DoEvents();
        }
}
```

Üçlüses dengesini ve zenginliğini koruyabilmek için, aynı cümlede aynı üçlüses birden fazla sayıda geçiyorsa, cümleye ceza puanı uygulandı. Normalde üçlüsesin frekans değeri cümlenin skoruna eklenmekteydi. Geliştirme ile, üçlüsesin frekans değerinin, *gecme_sayisi* değişkeninin değerine bölümü cümle skoruna eklendi. *gecme_sayisi* 1 ise, üçlüses frekansı cümle skoruna aynen eklenmekte; *gecme_sayisi* 2 ise üçlüses frekansının yarısı cümle skoruna eklenmekte…

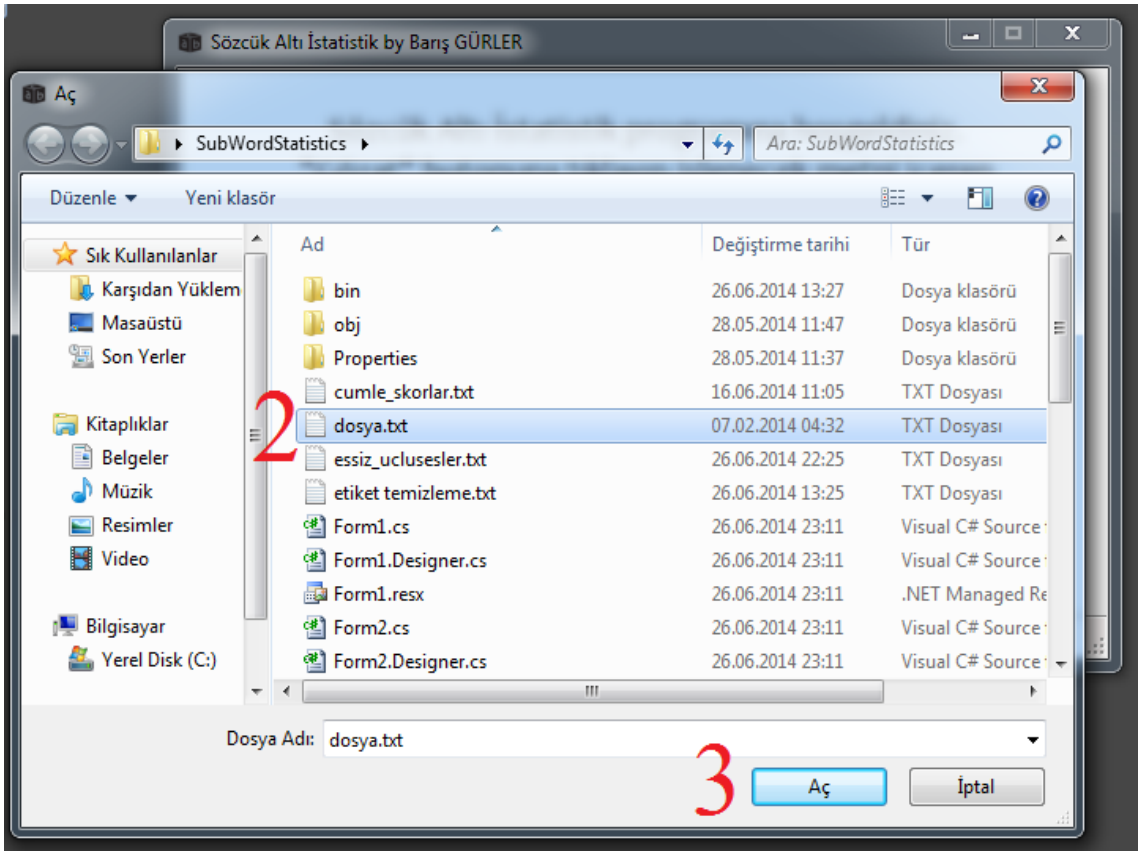
Bu işlemlerin herkes tarafından kolaylıkla yapılabilmesi için bir yazılım geliştirilmiştir. Yazılımın kodları C# diliyle yazılmış olup, çalıştırılabilmesi için bilgisayarda .NET Framework 4.0 yüklü olmalıdır.

Her adımda basılması gereken butonlar aktif, henüz basılmaması gereken butonlar pasif yapılmıştır. Bu şekilde yazılımın yanlış kullanımının önüne geçilmiştir. Programın kullanımı şu şekildedir:

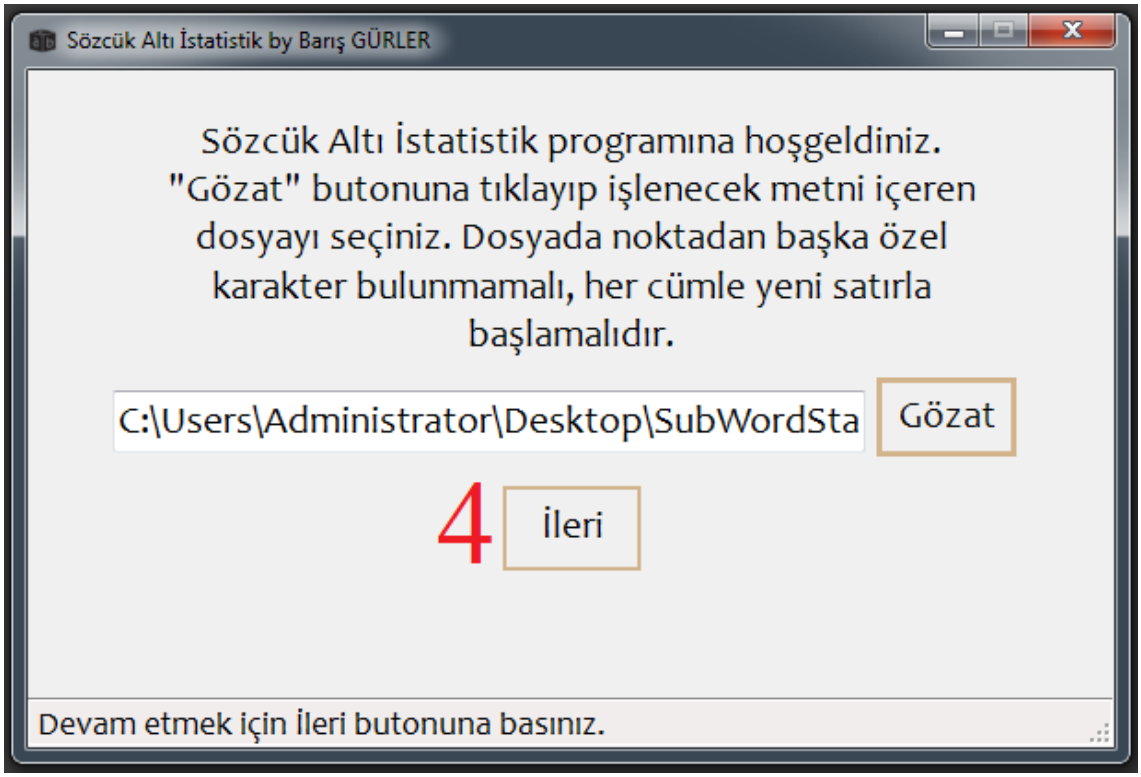
1. "Gözat" butonuna tıklanır (Resim 3.5 (a)).
2. İşlenecek metni içeren dosya seçilir (Resim 3.5 (b)). Dosyada noktadan başka özel karakter bulunmamalı ve her cümle yeni bir satırla başlamalıdır.
3. "Aç" butonuna tıklanır (Resim 3.5 (b)).
4. "İleri" butonuna tıklanır (Resim 3.5 (c)).
5. "Tüm Üçlüsesleri Bul" butonuna tıklanır (Resim 3.5 (d)).
6. İşlem (Resim 3.5 (e)) sona erdiğinde;
    a. Üretilen dosyayı görüntülemek istenirse "Dosyayı Göster" butonuna tıklanır (Resim 3.5 (f)).
    b. Bir sonraki adıma geçmek için "İleri" butonuna tıklanır (Resim 3.5 (f)).
7. "Tekrar Etmeyen Üçlüsesleri Bul" butonuna tıklanır (Resim 3.5 (g)).
8. "İleri" butonuna tıklanır (Resim 3.5 (h)).
9. "Frekansları Hesapla" butonuna tıklanır (Resim 3.5 (i)).
10. "İleri" butonuna tıklanır (Resim 3.5 (j)).
11. "Cümle Skorlarını Hesapla" butonuna tıklanır (Resim 3.5 (k)).
12. "Dosyayı Göster" butonuna tıklanır (Resim 3.5 (l)).



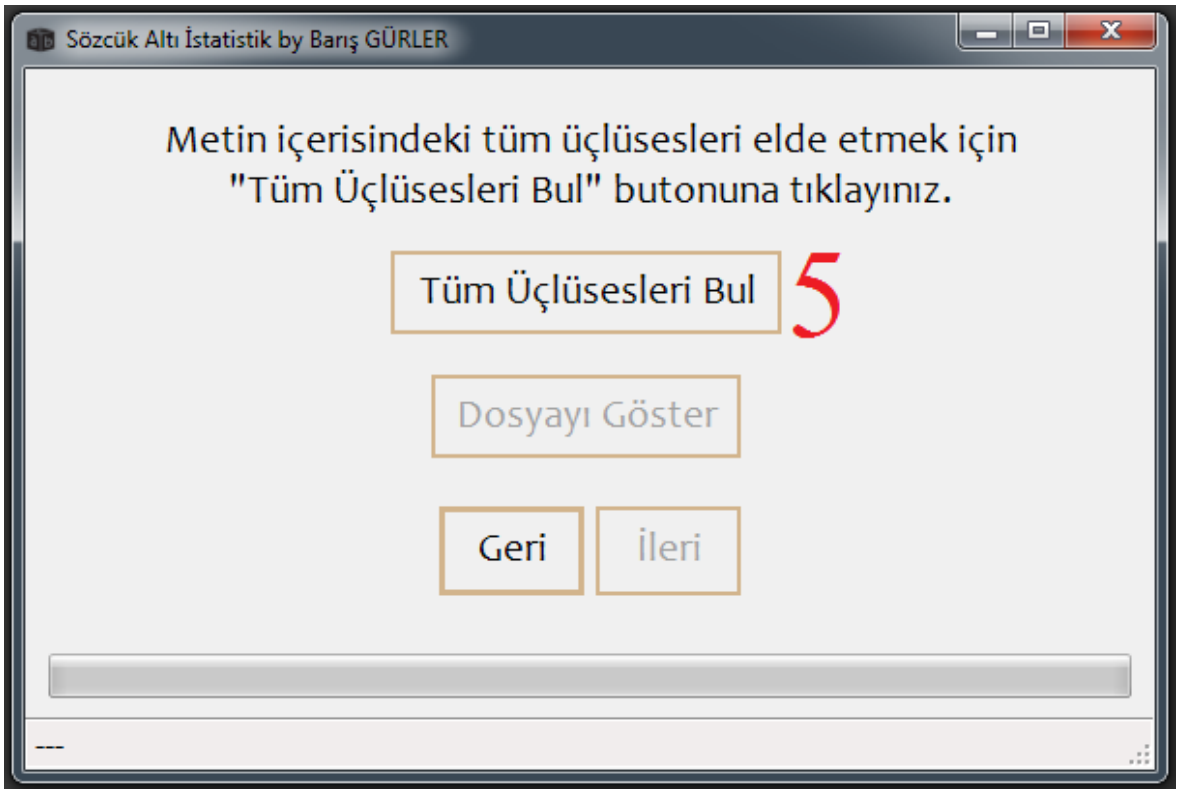
Resim 3.5 (a). Sözcük altı istatistik yazılımı (1)



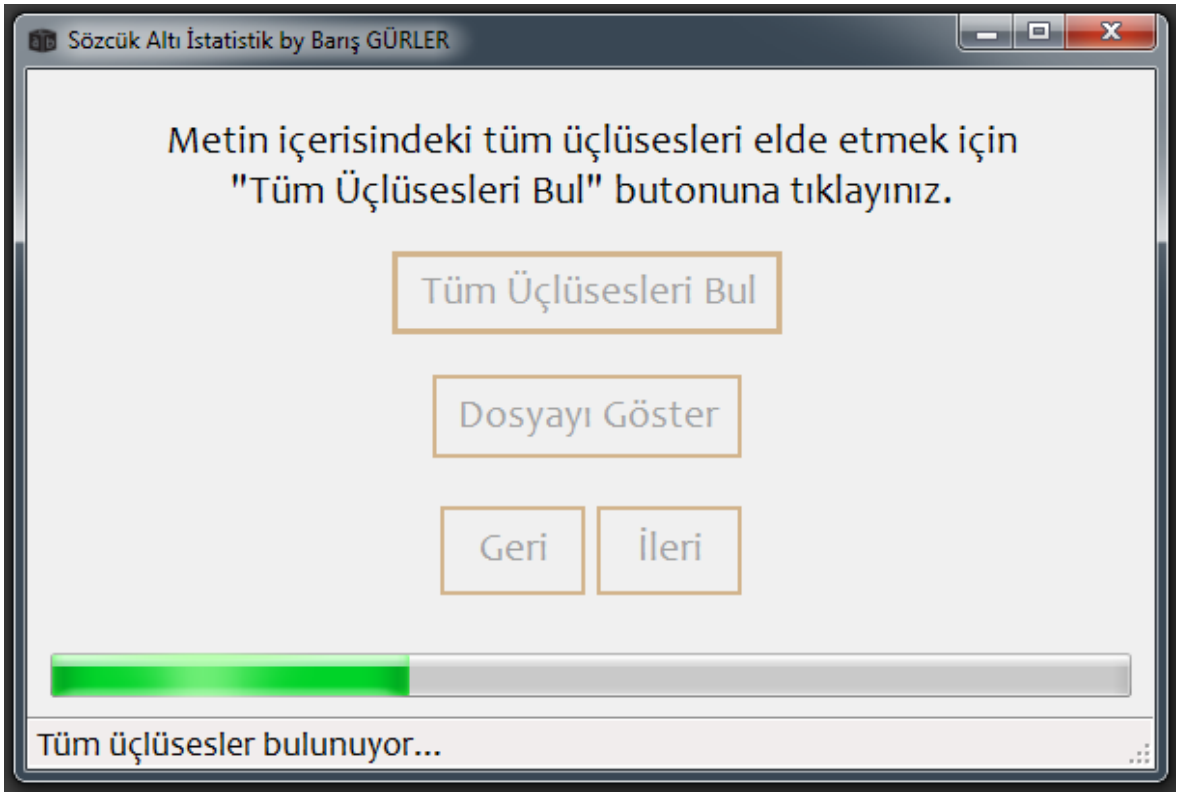
Resim 3.5 (b). Sözcük altı istatistik yazılımı (2-3)



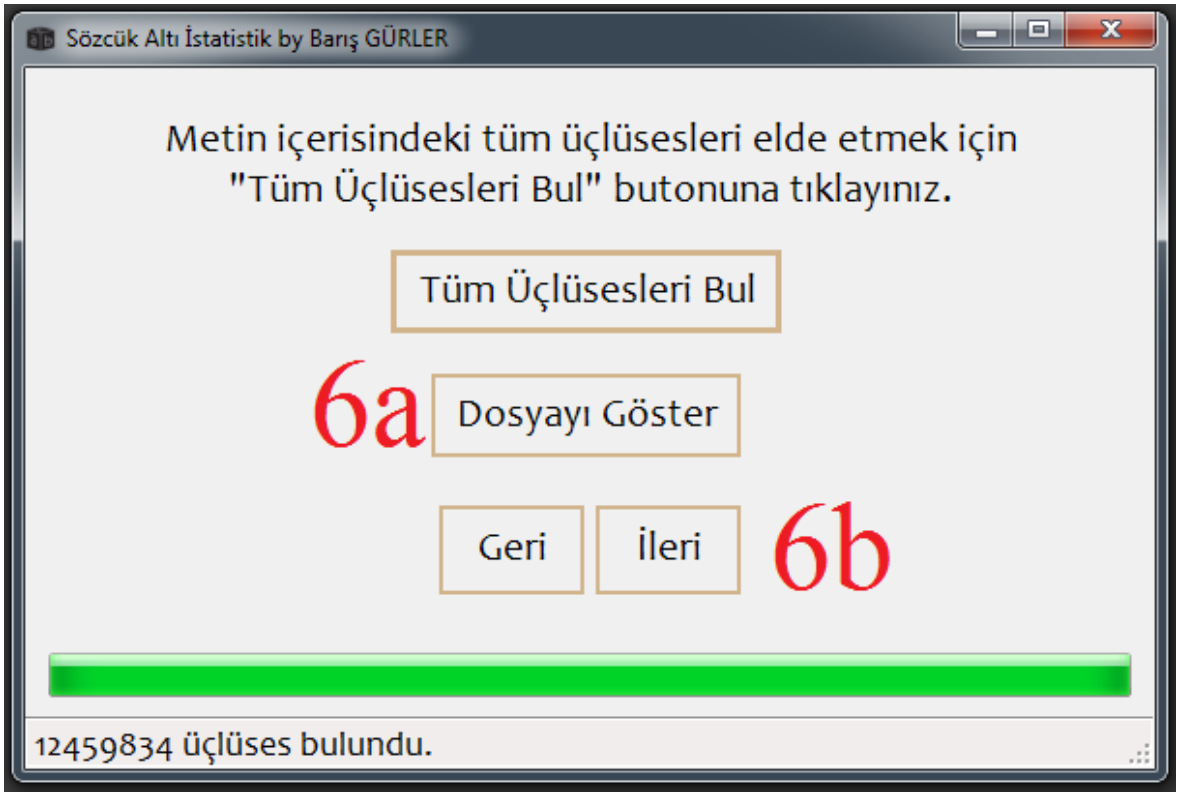
Resim 3.5 (c). Sözcük altı istatistik yazılımı (4)



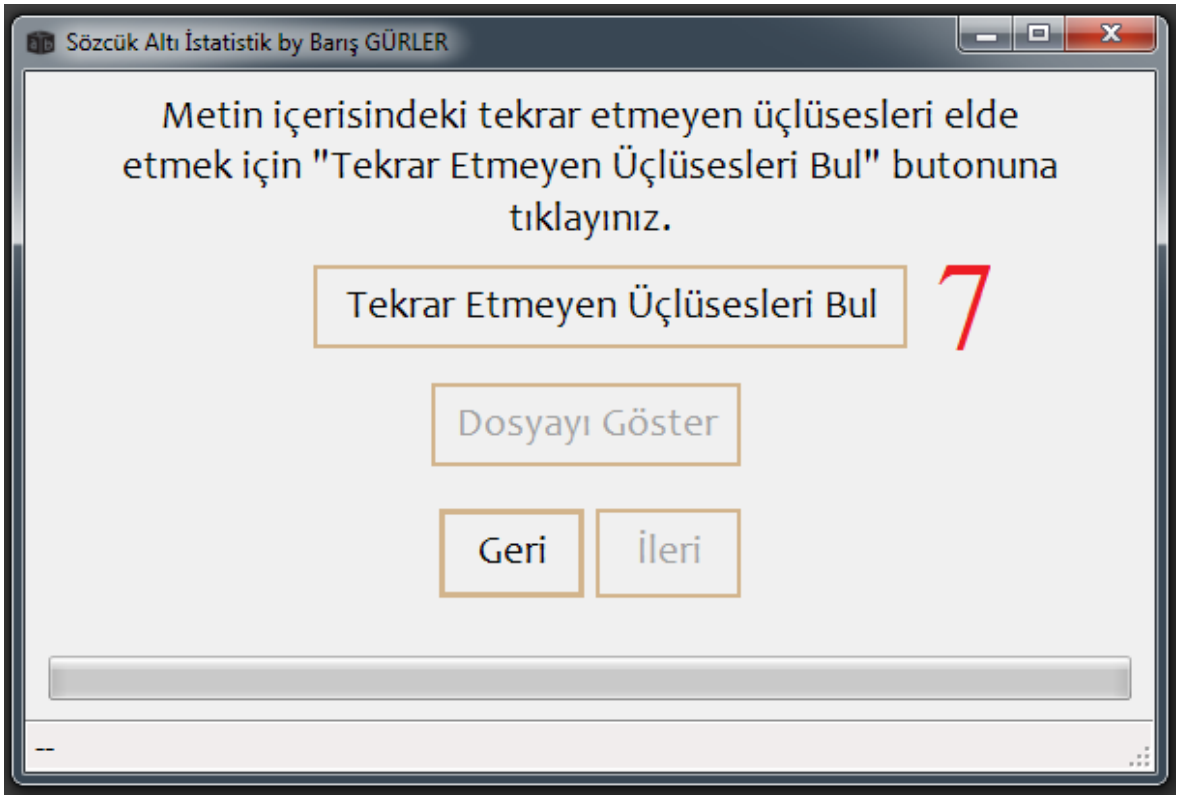
Resim 3.5 (d). Sözcük altı istatistik yazılımı (5)



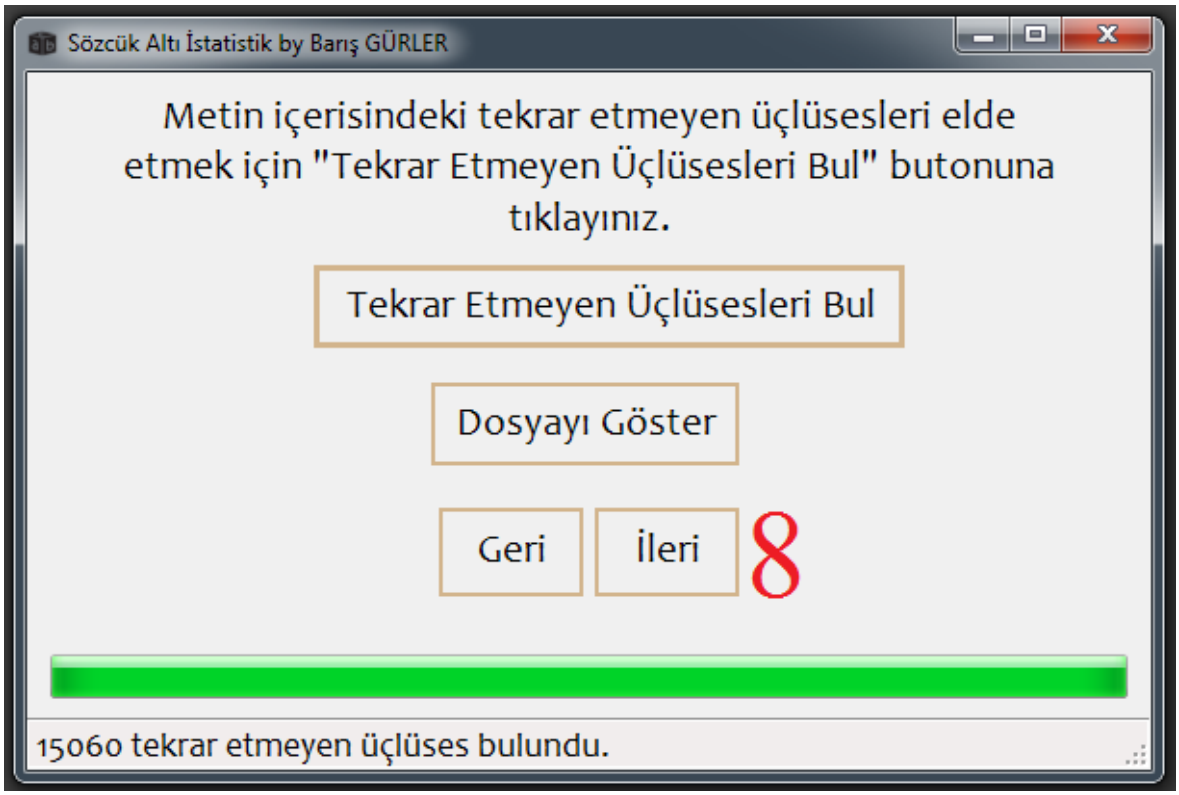
Resim 3.5 (e). Sözcük altı istatistik yazılımı (işlem)



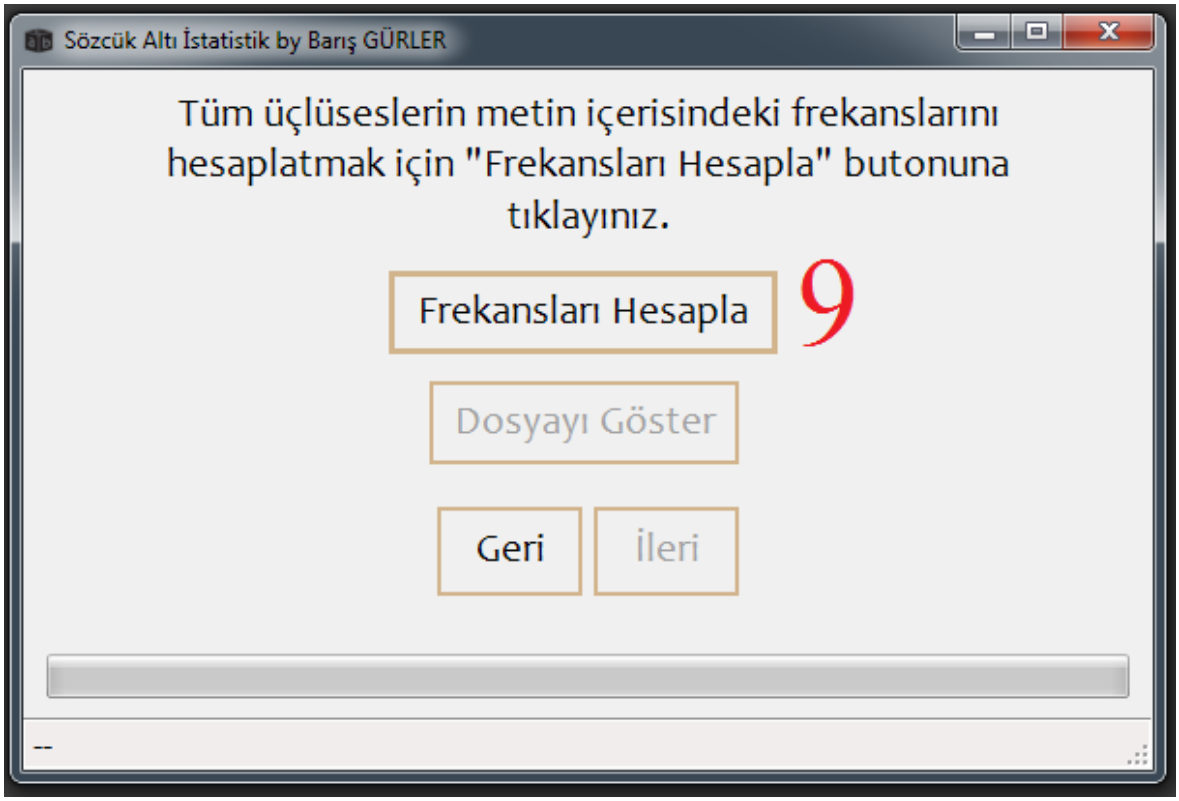
Resim 3.5 (f). Sözcük altı istatistik yazılımı (6a-6b)



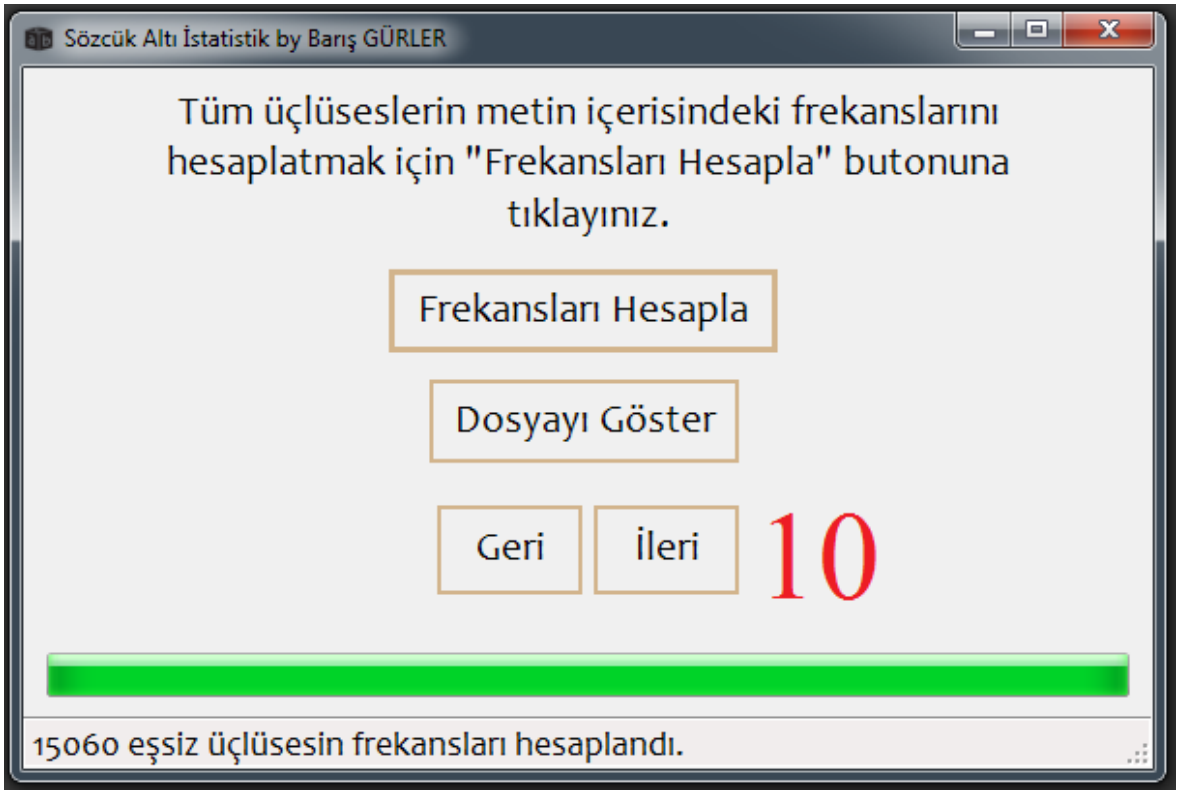
Resim 3.5 (g). Sözcük altı istatistik yazılımı (7)



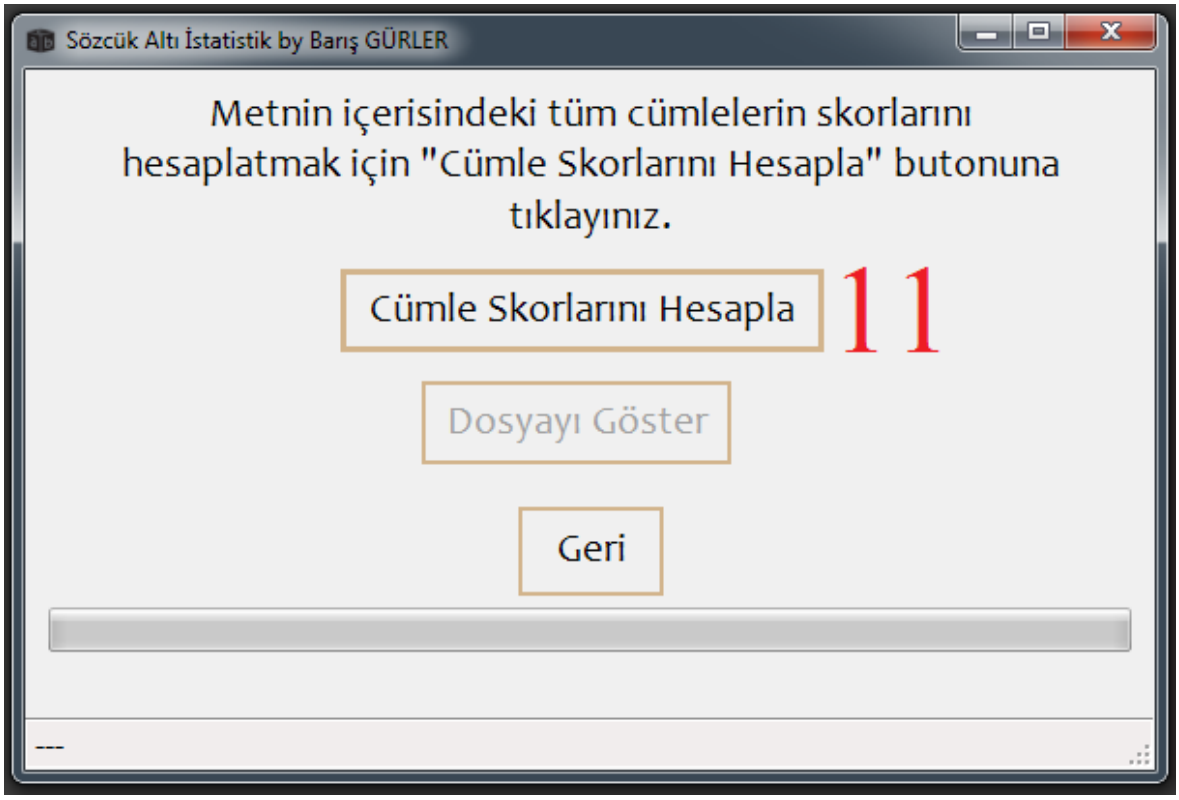
Resim 3.5 (h). Sözcük altı istatistik yazılımı (8)



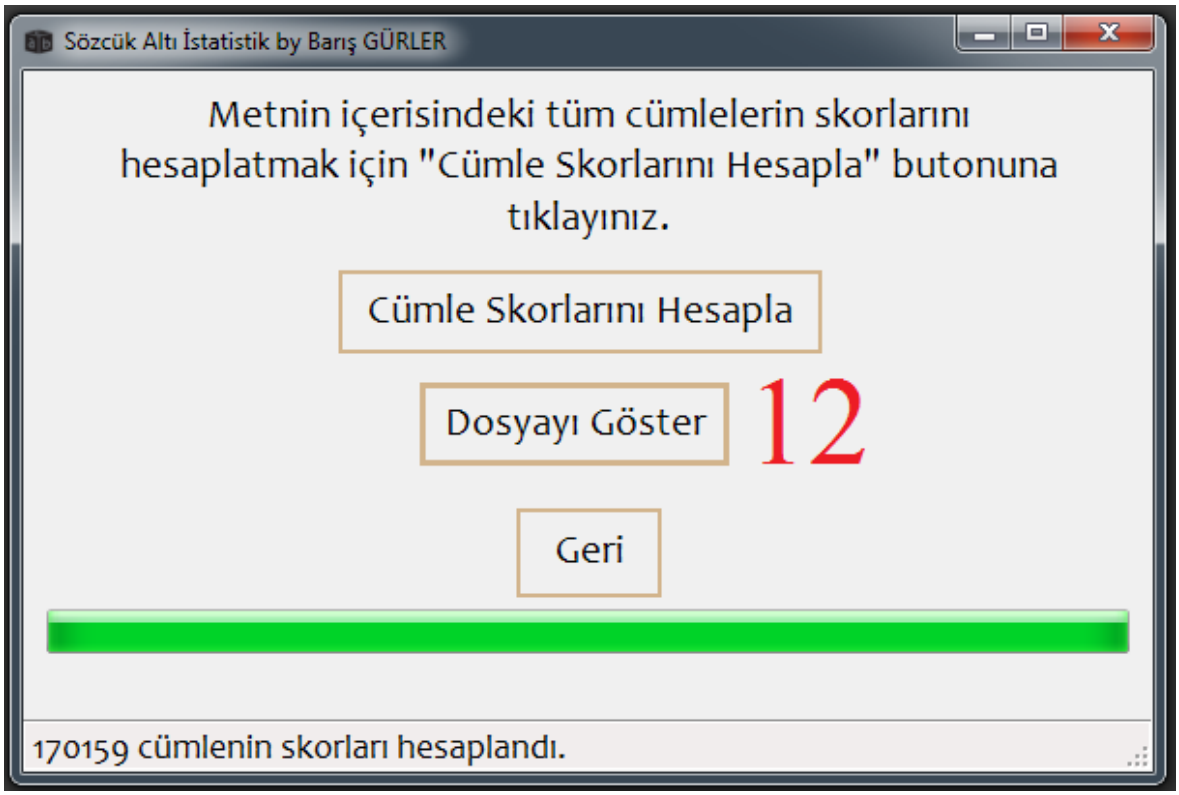
Resim 3.5 (i). Sözcük altı istatistik yazılımı (9)



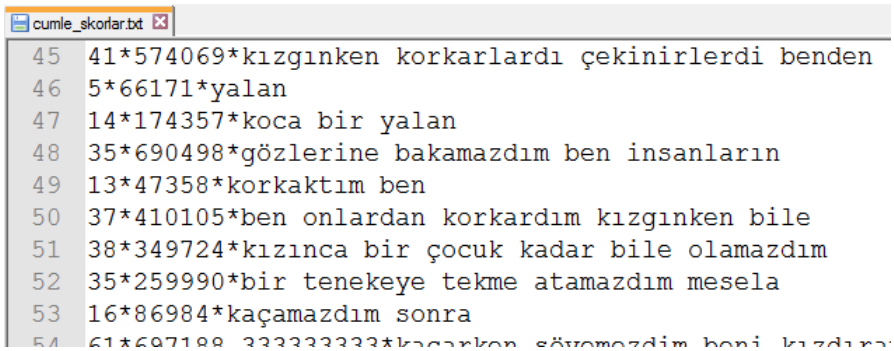
Resim 3.5 (j). Sözcük altı istatistik yazılımı (10)



Resim 3.5 (k). Sözcük altı istatistik yazılımı (11)



Resim 3.5 (l). Sözcük altı istatistik yazılımı (12)



Resim 3.6. Sözcük altı istatistik yazılımının çıktısı

9. ve 12. adımda üretilen dosyaların içeriği tablo biçimindedir. Üretilen metin dosyaları oldukça büyüktür. Programın çalışmasını yavaşlatmaması için çıktılar txt dosyaları halinde yapılmış, verilerin Excel programına kolayca aktarılabilmesi için sütunlar birbirinden *(yıldız) karakteriyle ayrılmıştır.

9. adımda üretilen dosyada (ucluses_frekanslar.txt) ilk sütun üçlüsesi, ikinci sütun ise o üçlüsesin metin içerisindeki frekansını ifade etmektedir.

12. adımda üretilen dosyada (cumle_skorlar.txt) ilk sütun cümlenin kaç karakterden oluştuğunu, ikinci sütun cümlenin skorunu, üçüncü sütun ise cümlenin kendisini ifade etmektedir. Dosya içeriği Resim 3.6'da verilmiştir.

Kullanılan algoritma sonucunda elde edilen en yüksek skor alan cümlelerin içerisinden, konuşmacılara okutmak üzere 60 tanesi seçilmiştir. Bu cümlelerin içerdiği üçlüseslerin frekansları Çizelge 3.6'da verilmiştir.

Çizelge 3.6. Seçilen 60 cümlenin üçlüses frekansları

Sıra	Üçlüses	Frekans	Sıra	Üçlüses	Frekans
1	l-a+r	85	21	a-n+l	16
2	l-e+r	68	22	n-i+n	16
3	e-r+i	66	23	n-b+i	15
4	a-r+ı	55	24	e-n+d	15
5	r-i+n	50	25	i-y+o	15
6	r-ı+n	40	26	a-l+a	13
7	i-n+i	37	27	a-y+a	12
8	b-i+r	34	28	e-n+i	12
9	n-d+a	27	29	r-l+e	12
10	y-o+r	26	30	n-d+i	12
11	i-l+e	24	31	k-e+n	11
12	ı-n+d	24	32	a-m+a	11
13	n-l+a	24	33	k-l+e	11
14	a-r+a	22	34	a-s+ı	10
15	n-d+e	20	35	i-n+e	10
16	k-l+a	18	36	e-d+i	10
17	d-e+n	18	37	d-a+n	10
18	ı-n+ı	16	38	a-r+d	10
19	i-n+d	16	39	a-n+ı	9
20	i-r+i	16	40	n-ı+n	9

2 milyon kelimelik metnin içerisinde en yüksek frekansa sahip olan 40 üçlüsesin, seçilen 60 cümledeki sırası Çizelge 3.7'de verilmiştir. e-s+i, i-l+i ve a-n+a üçlüsesleri dışında kalan üçlüsesler için sonuçların birbirine yakın olduğu gözlenmiştir.

Çizelge 3.7. Üçlüses frekans sıralama karşılaştırması

Üçlüses	Sıra (2M)	Sıra (60)	Üçlüses	Sıra (2M)	Sıra (60)
l-a+r	1	1	d-e+n	21	17
l-e+r	2	2	a-d+a	22	55
e-r+i	3	3	r-ı+n	23	6
b-i+r	4	8	e-d+i	24	36
a-r+ı	5	4	e-n+i	25	28
a-r+a	6	14	e-d+e	26	47
n-d+a	7	9	e-s+i	27	161
i-n+i	8	7	n-l+a	28	13
y-o+r	9	10	n-i+n	29	22
n-d+e	10	15	n-ı+n	30	40
ı-n+ı	11	18	i-y+e	31	68
r-i+n	12	5	i-n+e	32	35
a-s+ı	13	34	a-l+a	33	26
i-l+e	14	11	e-l+e	34	41
ı-n+d	15	12	d-a+n	35	37
i-n+d	16	19	i-l+i	36	108
a-y+a	17	27	s-ı+n	37	42
a-n+ı	18	39	e-r+e	38	56
a-m+a	19	32	o-l+a	39	48
l-a+n	20	67	a-n+a	40	162

2 milyon kelimelik metindeki en yüksek frekansa sahip olan 40 üçlüsesin, 60 cümle içerisindeki frekansları Çizelge 3.8'de verilmiştir.

Çizelge 3.8. Üçlüses frekans dağılımı

Üçlüses	Sıra (2M)	Frekans (60)	Üçlüses	Sıra (2M)	Frekans (60)
l-a+r	1	85	d-e+n	21	18
l-e+r	2	68	a-d+a	22	7
e-r+i	3	66	r-ı+n	23	40
b-i+r	4	34	e-d+i	24	10
a-r+ı	5	55	e-n+i	25	12
a-r+a	6	22	e-d+e	26	8
n-d+a	7	27	e-s+i	27	3
i-n+i	8	37	n-l+a	28	24
y-o+r	9	26	n-i+n	29	16
n-d+e	10	20	n-ı+n	30	9
ı-n+ı	11	16	i-y+e	31	6
r-i+n	12	50	i-n+e	32	10
a-s+ı	13	10	a-l+a	33	13
i-l+e	14	24	e-l+e	34	9
ı-n+d	15	24	d-a+n	35	10
i-n+d	16	16	i-l+i	36	4
a-y+a	17	12	s-ı+n	37	9
a-n+ı	18	9	e-r+e	38	7
a-m+a	19	11	o-l+a	39	8
l-a+n	20	6	a-n+a	40	3

OTD içindeki kelimelerin de skorları hesaplanmıştır. Bu tez çalışmasında bu veriye ihtiyaç duyulmasa dahi, başka araştırmalarda kullanılabilir. En yüksek skoru alan 40 kelime Çizelge 3.9'da verilmiştir.

Çizelge 3.9. En yüksek skor alan 40 kelime

Sıra	Kelime	Skor	Sıra	Kelime	Skor
1	pelerin	31387,857	21	bitleri	23954,571
2	keleri	30690,500	22	çoklarınca	23827,400
3	yuları	29564,500	23	bezeleri	23586,625
4	birler	29511,000	24	arınık	23207,333
5	çokları	28119,714	25	arınış	23000,000
6	erin	27886,250	26	rindane	22870,571
7	karıncıkları	26878,833	27	balerinlik	22711,400
8	karındaş	26451,750	28	kipleri	22628,857
9	raddelerinde	26339,333	29	yular	22622,000
10	balar	26083,400	30	sikalar	22531,571
11	larp	25694,500	31	bolarabilme	22456,091
12	bilar	25400,800	32	verile	22454,000
13	akarlar	24807,286	33	akkarıncalar	22367,000
14	geriniş	24702,857	34	alarmak	22316,429
15	minderi	24700,857	35	karınlama	22304,667
16	polarıcı	24625,375	36	yerinel	22282,000
17	yerindelik	24403,700	37	kararış	22247,143
18	flamanlar	24396,111	38	taraklılar	22183,600
19	kayaçları	24208,667	39	alarga	22140,500
20	fundalar	24161,000	40	denden	21781,833

Bu kelimeler içerisinde Türkçe'de çok az kullanılan kelimeler de bulunmaktadır. Burada hesaplanan, kelimelerin frekansından ziyade, içerdikleri üçlüseslerin OTD içerisinde geçme sayılarının toplamıdır.

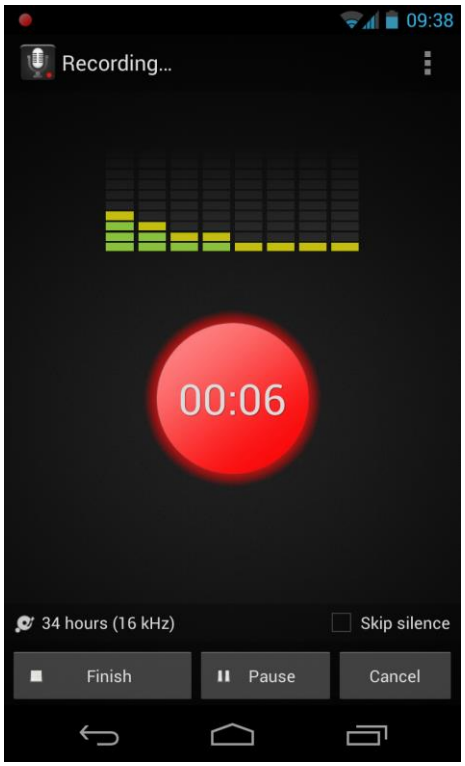
Konuşmacılara okutmak amacıyla bu listenin en üstünde yer alan cümleler içerisinden 60 tanesi seçildi. Seçilen bu cümleler 30 konuşmacıya okutuldu ve 1800 adet ses dosyası elde edildi.

Çalışmanın hedef kitlesi, 89-95 yılları arasında doğmuş olan üniversite öğrencileri ile sınırlandırıldı.

Ses kayıtları; gürültüsüz sayılabilecek bir ortamda, Samsung Note 2 model bir telefonla yapılmıştır. Ses kayıt yazılımı olarak Smart Voice Recorder [18] isimli ücretsiz android uygulaması kullanılmıştır. Kayıt yapmak için farklı bir araç ve yazılım da kullanılabilir.

Ses kayıtları 16 KHz frekansta, wav formatında alınmıştır. HTK (Hidden Markov Model Toolkit) [19]'nın bu özelliklere sahip olan ses dosyalarını desteklediği The HTK Book (3.4 sürümü için) [20] isimli başvuru kitabının 94. sayfasında belirtilmiştir.

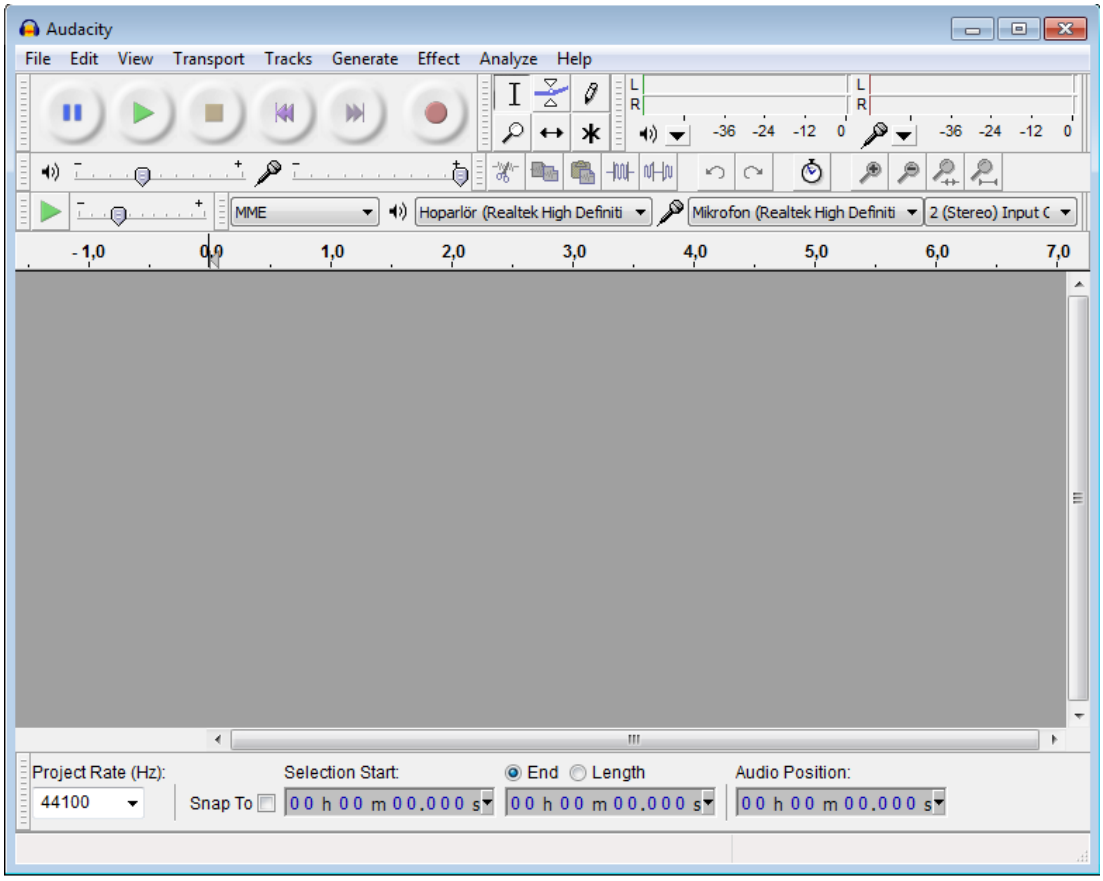
Smart Voice Recorder (Resim 3.7) yazılımıyla 16 KHz wav formatında kayıt yapmak için herhangi bir ayar yapmaya gerek yoktur. Yazılımın varsayılan ayarı bu şekildedir.



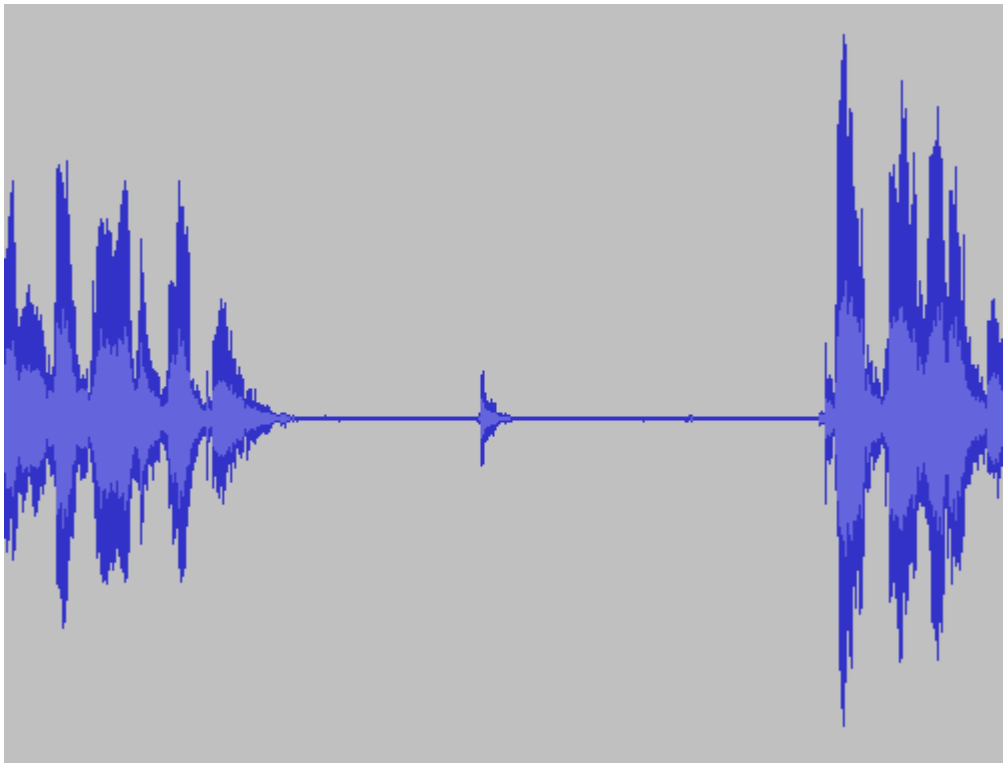
Resim 3.7. Smart Voice Recorder yazılımı

Her konuşmacıya 60 cümle okutuldu ve 60 adet ses dosyası elde edildi. Daha sonra her dosyanın içerisindeki 60 cümleye tek tek gürültü azaltma uygulandı ve her biri ayrı dosyalara çıkartıldı. Oluşan dosyaların isimlendirilmesinde, kxxcxx.wav formatı benimsendi. Burada kxx ile hangi konuşmacı olduğu, cxx ile de hangi cümlenin söylendiği belirtilmeye çalışıldı. Örneğin, "k05c26.wav" dosyası, 26 numaralı cümleyi 5. konuşmacının seslendirdiği örneği nitelemektedir.

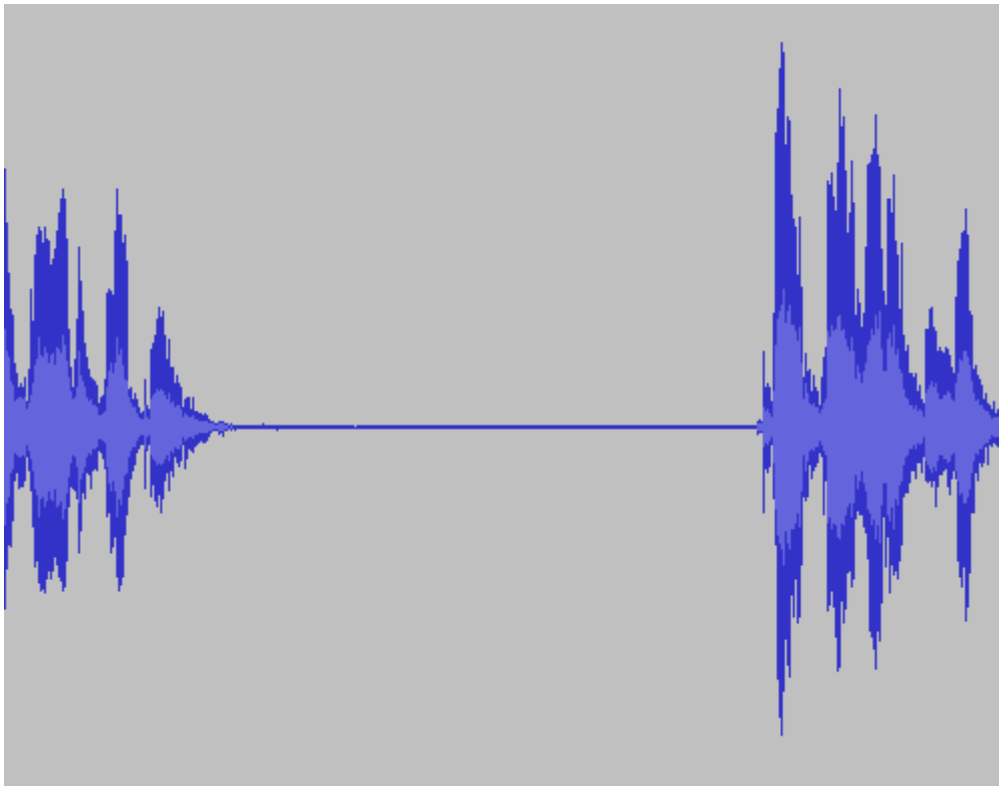
Gürültü azaltma (Resim 3.9 (a-b)) ve cümlelerin ayrılması işlemleri Audacity (Resim 3.8) [21] isimli ücretsiz programla yapılmıştır.



Resim 3.8. Audacity yazılımı

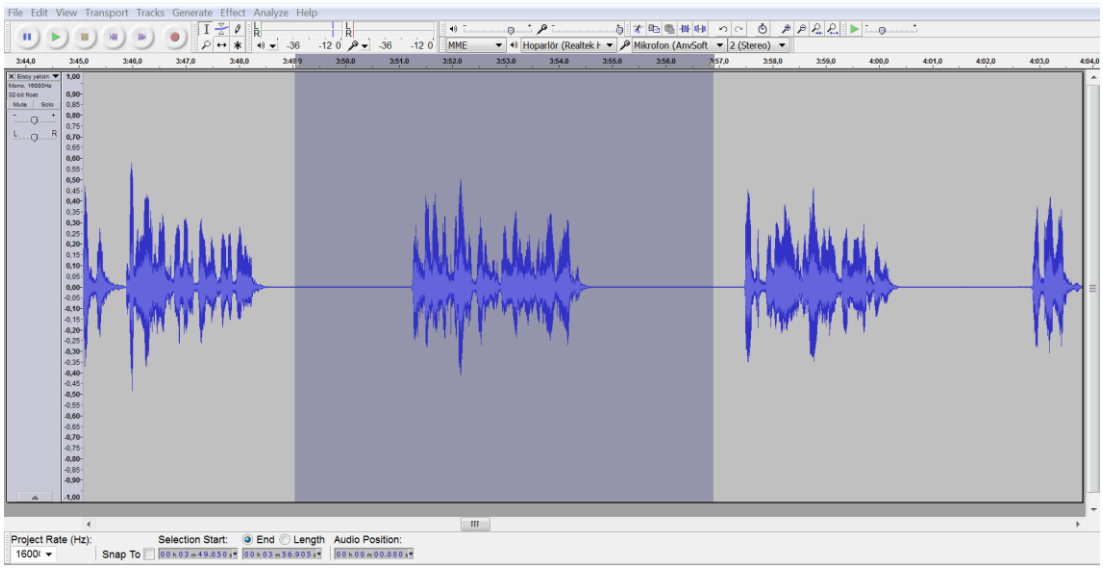


Resim 3.9 (a). Ses dosyasındaki gürültünün giderilmesi (1)

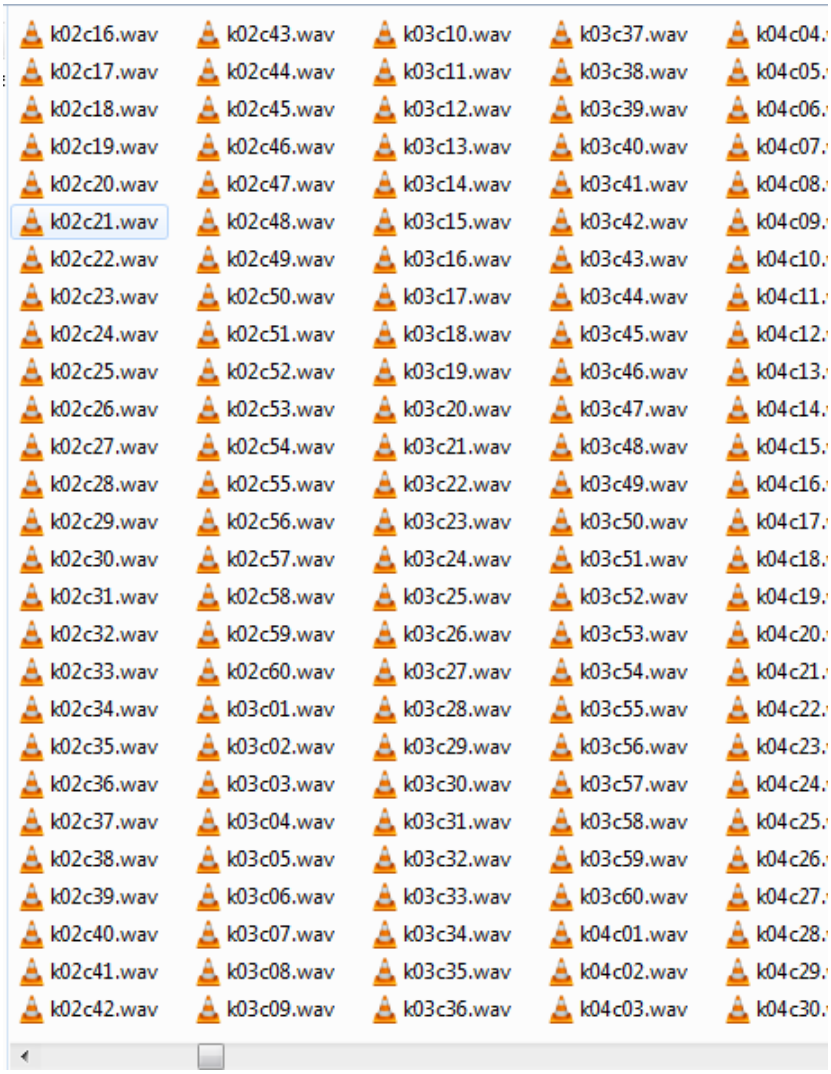


Resim 3.9 (b). Ses dosyasındaki gürültünün giderilmesi (2)

Gürültü azaltma işleminden sonra, her cümlenin okunduğu kısım seçilerek (Resim 3.10) ayrı bir dosyaya çıkarıldı (Resim 3.11). “sil“ foneminin sistem tarafından tanınabilmesi için okuma yapılan kısım, önünde ve arkasında bir miktar sessizlik ile beraber seçilmiştir.



Resim 3.10. Cümlelerin ses dosyasından ayrılması



Resim 3.11. Ses dosyaları

Bu işlemlerin ardından, fonem seviyesindeki zaman damgalarının üretilmesi için HTK aracındaki yazılımların kullanılması gerekti [22].

HTK'nın Windows üzerinde çalışmaya hazır, derlenmiş binary kodu resmi olarak sunulmamaktadır. Yazılımın kaynak kodu ücretsiz olarak edinilebilmektedir [23].

Kaynak kodunun Windows üzerinde derlenebilmesi için Microsoft Visual Studio programının kurulu olması gerekmektedir. Aşağıda kaynak kodun derlenmesi için yapılması gereken işlemler açıklanmıştır [24].

Öncelikle indirilen zip dosyasındaki dosyalar çıkarılır ve komut satırında sırayla şu komutlar çalıştırılır:

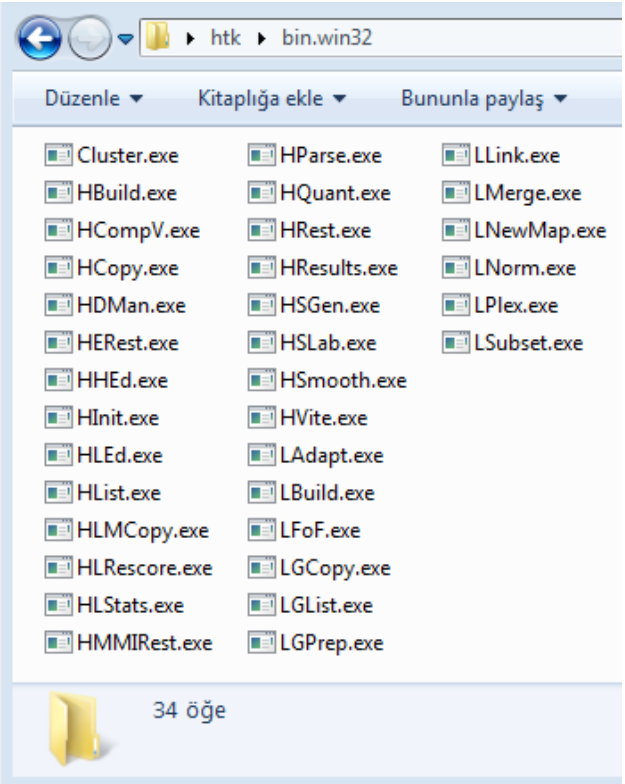
- cd htk
- mkdir bin.win32
- VCVARS32

Ardından, HTK araçlarına (HTKTools) işlevselliğini veren HTK kütüphanesi (HTKLib) oluşturulur:

- cd HTKLib
- nmake /f htk_htklib_nt.mkf all
- cd ..

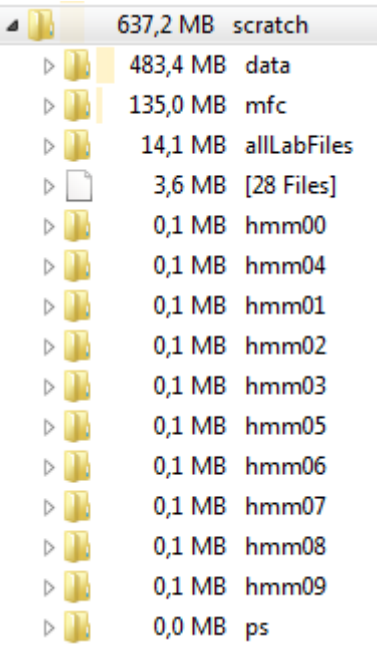
Son işlem olarak HTKTools oluşturulur ve derlenmiş HTK dosyaları elde edilir (Resim 3.12):

- cd HTKTools
- nmake /f htk_htktools_nt.mkf all
- cd ..
- cd HLMLib
- nmake /f htk_hlmlib_nt.mkf all
- cd ..
- cd HLMTools
- nmake /f htk_hlmtools_nt.mkf all
- cd ..



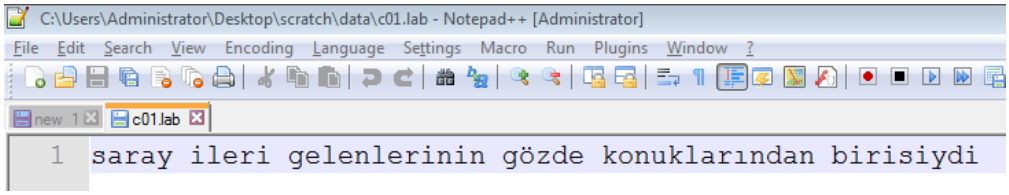
Resim 3.12. Derlenmiş HTK dosyaları

Çalışma sonucunda oluşacak tüm dosyalar scratch adındaki tek bir dizin altında toplanmıştır (Resim 3.13). Gerekli alt dizinler oluşturulmuştur.



Resim 3.13. Scratch dizininin yapısı

Ses dosyaları (wav) ve bunların içerdikleri cümlelerin yazılışını (transkript) içeren ".lab" uzantılı dosyalar "data" dizininde yer almaktadır (Resim 3.14).



Resim 3.14. c01.lab dosyasının içeriği

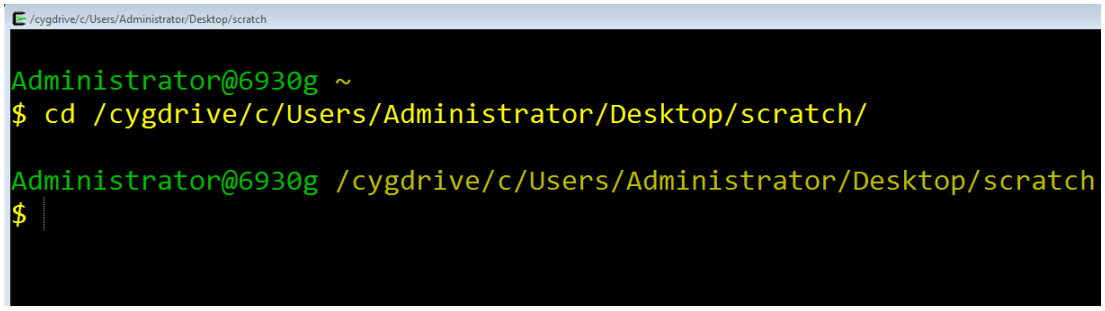
Ses dosyalarından elde edilecek olan MFCC (Mel-frequency cepstral coefficients) öznitelik vektörleri için "mfc" isimli bir dizin oluşturulmuştur.

Sistem 9 adımda eğitileceği için "hmm00 – hmm09" dizinleri oluşturulmuştur.

Bazı işlemlerin kolaylaştırılması için kullanılacak olan perl scriptleri için "ps" dizini oluşturulmuştur.

Derlenmiş HTK dosyaları bin.win32 dizinine çıkartılmıştı. Bunların içindeki "HCompV.exe, HCopy.exe, HDMan.exe, HERest.exe, HHEd.exe, HLEd.exe, HVite.exe" isimli dosyalar scratch dizinine kopyalandı.

Perl scriptleri ve bazı unix komutları kullanılacağı için cygwin (Resim 3.15) isimli yazılımın kurulması gerekmektedir.

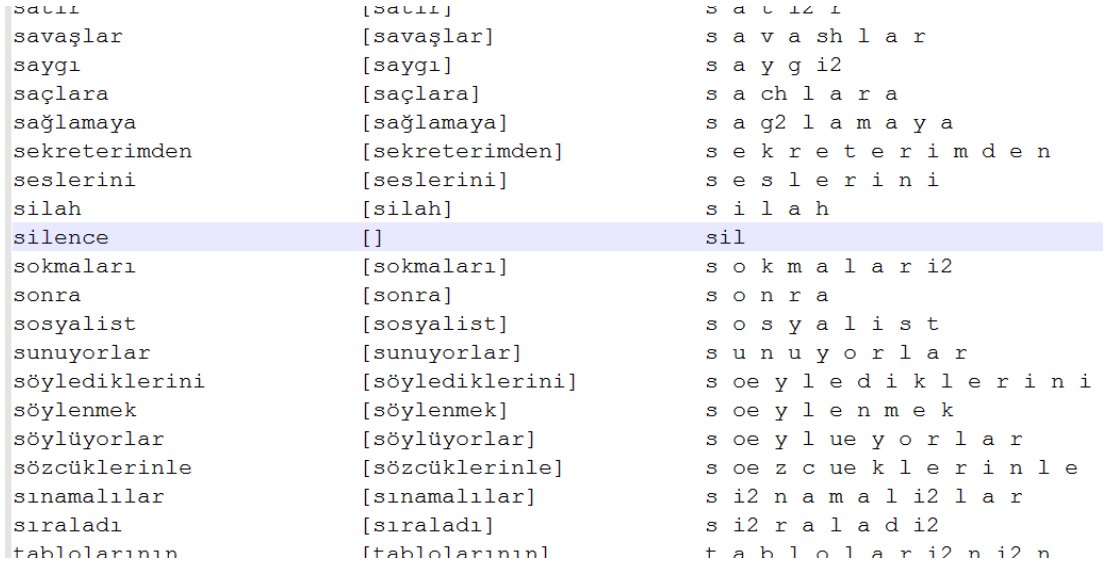


Resim 3.15. Cygwin programı

American Standard Code for Information Interchange (ASCII) setinde olmayan harfler (ç, ğ, ı, ş, ö, ü) için yeni değerler tanımlandı (ch, g2, i2, sh, oe, ue).

- ç → ch
- ğ → g2
- ı → i2
- ş → sh
- ö → oe
- ü → ue

Telaffuz sözlüğü (pDict.txt) oluşturuldu (Resim 3.16). Bu dosya, seslendirilen cümlelerdeki tüm kelimeleri ve bunların telaffuzlarını içermek zorundadır. En soldaki sütun, kelimeleri içermektedir. En sağda bu kelimelerin fonetik açılımı, ortada ise ürettiği semboller yer almaktadır. Örneğin "silence", cümlenin başındaki ve sonundaki sessizliği ifade ettiğinden, herhangi bir sembol üretmemektedir.



savaşlar	[savaşlar]	s a v a sh l a r
saygı	[saygı]	s a y g i2
saçlara	[saçlara]	s a ch l a r a
sağlamaya	[sağlamaya]	s a g2 l a m a y a
sekreterimden	[sekreterimden]	s e k r e t e r i m d e n
seslerini	[seslerini]	s e s l e r i n i
silah	[silah]	s i l a h
silence	[]	sil
sokmaları	[sokmaları]	s o k m a l a r i2
sonra	[sonra]	s o n r a
sosyalist	[sosyalist]	s o s y a l i s t
sunuyorlar	[sunuyorlar]	s u n u y o r l a r
söylediklerini	[söylediklerini]	s oe y l e d i k l e r i n i
söylenmek	[söylenmek]	s oe y l e n m e k
söylüyorlar	[söylüyorlar]	s oe y l ue y o r l a r
sözcüklerinle	[sözcüklerinle]	s oe z c ue k l e r i n l e
sınamalılar	[sınamalılar]	s i2 n a m a l i2 l a r
sıraladı	[sıraladı]	s i2 r a l a d i2

Resim 3.16. Telaffuz sözlüğünün görünümü

Kelime listesini içeren dosya (word.list) oluşturuldu (Resim 3.17). Seslendirilen cümlelerdeki tüm kelimeler bu dosyadadır.

word.list

```
61 birilerinin
62 birinde
63 birindeydi
64 birini
65 birisi
66 birisiydi
67 boş
68 bu
69 buldu
70 bunlar
71 bunları
72 bunların
```

Resim 3.17. word.list dosyasının görünümü

Bu hazırlıklar tamamlandıktan sonra ilk iş olarak fonemlerin listesini içeren dosya (phone1.list) oluşturuldu. Bunun için HDMan isimli program kullanıldı:

- HDMan.exe -m -w word.list -n phone1.list -l dlog -C myconfig dict.list pDict.txt

Bu komut word.list, myconfig ve pDict.txt isimli dosyaları girdi olarak alıp phone1.list, dlog ve dict.list isimli dosyaları üretmektedir. Çeşitli ayarların yapılmasını sağlayan myconfig dosyasının içeriği şöyledir:

- HSHELL: NONUMESCAPES = T

Üretilen phone1.list dosyasında sp fonemi yoksa eklenmelidir. Daha sonra bu dosyanın bir kopyası (phone0.list) oluşturulmalı, ve bu dosyadan sp fonemi silinmelidir (Resim 3.18).

phone1.list

```
14 s
15 y
16 o
17 u
18 sh
19 v
20 ch
21 g2
22 z
23 h
24 g
25 ue
26 f
27 oe
28 p
29 sp
30 sil
31
```

phone0.list

```
13 m
14 s
15 y
16 o
17 u
18 sh
19 v
20 ch
21 g2
22 z
23 h
24 g
25 ue
26 f
27 oe
28 p
29 sil
30
```

Resim 3.18. Fonem listesini içeren dosyaların görünümü

Lab dosyalarındaki verilerin tümünü içeren dosyayı (word0.mlf) oluşturmak için label2mlf isimli script kullanıldı (Resim 3.52):

- perl ps/label2mlf data/*.lab > word0.mlf

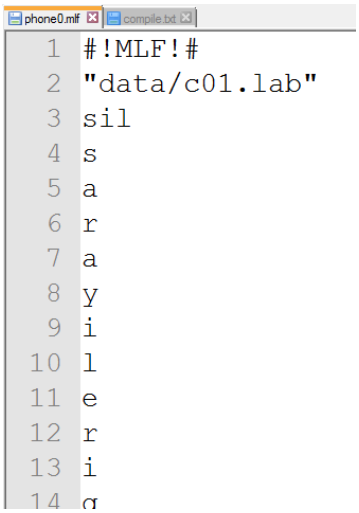
word0.mlf

```
1  #!MLF!#
2  "data/c01.lab"
3  saray
4  ileri
5  gelenlerinin
6  gözde
7  konuklarından
8  birisiydi
9  .
10 "data/c02.lab"
11 ayrıca
12 kendilerinden
13 olmayanları
```

Resim 3.19. mlf dosyasının görünümü

Benzer işlemin tüm fonemler için yapılmasıyla phone0.mlf dosyası oluşturuldu (Resim 3.53). Bu işlem, HTK araçlarından biri olan HLed ile kolay bir şekilde yapılabilir:

- HLed.exe -l 'data/' -d dict.list -i phone0.mlf mkPhones0.led word0.mlf



Resim 3.20. phone0.mlf dosyasının görünümü

Sonraki adım, ses dosyalarının MFCC öznitelik vektörlerinin çıkarılmasıdır. Bunun için öncelikle copy.cfg isimli kondigürasyon dosyası oluşturuldu. Bu dosyada yapılan ayarlar şöyledir:

- SOURCEKIND = WAVEFORM
- SOURCEFORMAT = WAVE
- SOURCERATE = 625.0 # 16KHz (100 ns)
- TARGETKIND = MFCC_0_D_A
- TARGETRATE = 100000.0 # 10 ms
- SAVECOMPRESSED = T
- SAVEWITHCRC = T
- WINDOWSIZE = 250000.0 # 25 ms
- USEHAMMING = T
- PREEMCOEF = 0.97
- NUMCHANS = 20
- CEPLIFTER = 22
- NUMCEPS = 12
- ENORMALISE = T
- NATURALREADORDER = T

Ses kayıtları wav formatında olduğundan, SOURCEKIND ve SOURCEFORMAT parametreleri buna uygun olarak seçildi.

Ses kayıtları 16 KHz örnekleme frekansına sahiptir. SOURCERATE parametresi KHz cinsinden değil, ns cinsinden değer beklemektedir. Bu nedenle:
1 / 16000 * 10000000 işlemi yapılarak 16 KHz değeri 625 ns'ye dönüştürülmüştür.

NUMCEPS parametresine 12 değeri verilerek, her çerçevede 12 MFCC katsayısı [c1,…, c12] olması sağlanmıştır.

TARGETKIND parametresinde:

- _0 eki ile c0 isimli null MFCC katsayısı tanımlanmıştır.
- _D eki ile [c0, c1,…, c12] katsayılarının delta katsayıları (birinci derece türevi) eklenmiştir.
- _A eki ile [c0, c1,…, c12] katsayılarının delta katsayıları (ikinci derece türevi) eklenmiştir.

SAVECOMPRESSED parametresine T (true) değeri verilerek, üretilecek MFCC dosyalarının sıkıştırılması sağlanmıştır.

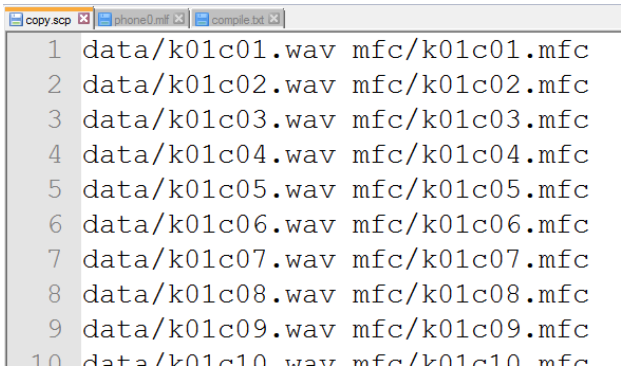
SAVEWITHCRC parametresine T değeri verilerek, üretilecek MFCC dosyalarına checksum bilgisi eklenmiştir.

USEHAMMING parametresine T değeri verilerek hamming pencereleme yöntemi kullanılmış, WINDOWSIZE parametresiyle de hamming pencerelerinin boyutu belirlenmiştir.

Diğer parametler için de HTKBook isimli başvuru kitabında verilen değerler kullanılmıştır.

Hcopy aracı ile MFCC elde etme işlemine başlamadan önce her wav dosyasının ve bunlara karşılık gelecek olan mfc dosyasının yolunu gösteren copy.scp dosyası (Resim 3.54) oluşturulmalıdır:

- perl ps/copyList > copy.scp



Resim 3.21. copy.scp dosyasının görünümü

Bu işlemin ardından HCopy aracı ile MFCC elde etme işlemi başlatıldı:

- HCopy.exe -T 1 -C copy.cfg -S copy.scp

3 durumlu SMM'lerin oluşturulması işlemine geçildi. Copy.cfg dosyasındaki TARGETKIND parametresine uygun olmak şartıyla bir prototip (proto) dosyası hazırlandı:

- MFCC_0: 13 vektör
- D (delta katsayıları): +13 vektör
- A (hızlandırma katsayıları): +13 vektör

Prototip dosyasının içeriği şöyledir:

~o <VecSize> 39 <MFCC_0_D_A>

~h "proto"

<BeginHMM>

<NumStates> 5

<State> 2

<Mean> 39

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

<Variance> 39

1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0

<State> 3

<Mean> 39

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

<Variance> 39

1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0

<State> 4

<Mean> 39

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

<Variance> 39

1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0

<TransP> 5

0.0 1.0 0.0 0.0 0.0

0.0 0.6 0.4 0.0 0.0

0.0 0.0 0.6 0.4 0.0

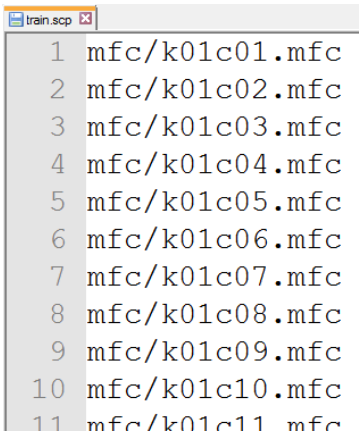
0.0 0.0 0.0 0.7 0.3

0.0 0.0 0.0 0.0 0.0

<EndHMM>

Sistemin eğitimi için gereken bilgilerin yer aldığı mfc dosyalarının yollarını içeren train.scp dosyası (Resim 3.22) oluşturuldu:

- perl ps/compList > train.scp



Resim 3.22. train.scp dosyasının görünümü

Bu işlemin ayrıca comp.cfg isimli ayrı bir ayar dosyasına (Resim 3.23) ihtiyacı vardır:

- HCompV.exe -C comp.cfg -f 0.01 -S train.scp -M hmm00 proto

comp.cfg

```
1  TARGETKIND = MFCC_0_D_A
2  TARGETRATE = 100000.0
3  SAVECOMPRESSED = T
4  SAVEWITHCRC = T
5  WINDOWSIZE = 250000.0
6  USEHAMMING = T
7  PREEMCOEF = 0.97
8  NUMCHANS = 20
9  CEPLIFTER = 22
10 NUMCEPS = 12
```

Resim 3.23. comp.cfg dosyası

Bu teklises (monophone) başlangıç SMM'lerini yeniden eğitmek için öncelikle macros dosyası, SMM tanımları (hmmdefs) ve cümle sırasına göre fonem listesi (phone0.mlf) oluşturulur:

- cat mactmp.txt hmm00/vFloors > hmm00/macros
- perl ps/init phone0.list > hmm00/hmmdefs
- perl ps/phone2label phone0.mlf

Eğitim işlemi için HERest aracı kullanılır:

- ./HERest.exe -A -D -T 1 -C comp.cfg -I phone0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm00/macros -H hmm00/hmmdefs -M hmm01 phone0.list

İşlem sona erdiğinde şöyle bir mesaj alınmaktadır:

Saving hmm's to dir hmm01

Reestimation complete - average log prob per frame = -7.761093e+001

- total frames seen = 1.567937e+006

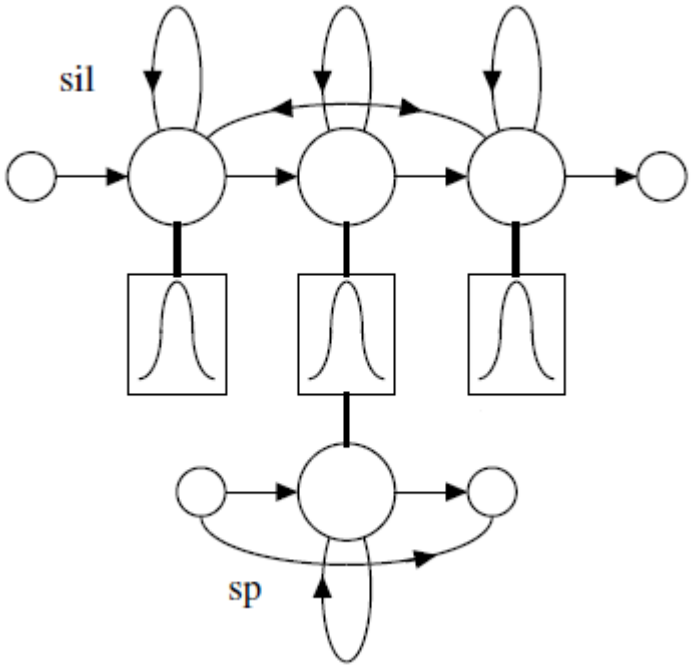
Ardından aynı işlem 2 defa daha (kaynak ve hedef dizinler değiştirilerek) yapılır:

- ./HERest.exe -A -D -T 1 -C comp.cfg -I phone0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm01/macros -H hmm01/hmmdefs -M hmm02 phone0.list

- ./HERest.exe -A -D -T 1 -C comp.cfg -I phone0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm02/macros -H hmm02/hmmdefs -M hmm03 phone0.list

Bu adıma kadar yapılan işlemlerde sp fonemi oluşturulmamıştı. Bu adımda ise, sil fonemi kullanılarak sistemin sp fonemini öğrenmesi sağlanmalıdır. Şekil 3.1'de görüleceği gibi, sp modeli 3 durumludur ve bunlardan sadece biri çıktı üretmektedir. Bu durum da 5 durumlu sil modelinin 3 numaralı durumuna bağlanmıştır (tied – state):

- perl ps/sil2sp hmm03/hmmdefs > hmm04/hmmdefs
- cp hmm03/macros hmm04/macros
- HHEd.exe -H hmm04/macros -H hmm04/hmmdefs -M hmm05 sil.hed phone1.list
- HLed -l 'data/' -d dict.list -i phone1.mlf mkPhones1.led word0.mlf
- perl ps/phone2label phone1.mlf



Şekil 3.1. sp modelinin oluşturulması [20]

Bu işlemler tamamlandıktan sonra, önceki eğitim adımları tekrarlanır:

- ./HERest.exe -A -D -T 1 -C comp.cfg -I phone1.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm05/macros -H hmm05/hmmdefs -M hmm06 phone1.list
- ./HERest.exe -A -D -T 1 -C comp.cfg -I phone1.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm06/macros -H hmm06/hmmdefs -M hmm07 phone1.list

Bundan sonraki işlemler verilerin yeniden hizalanmasıyla ilgilidir. Öncelikle dict.list dosyasına sil modeli için bir telaffuz bilgisi eklenir:

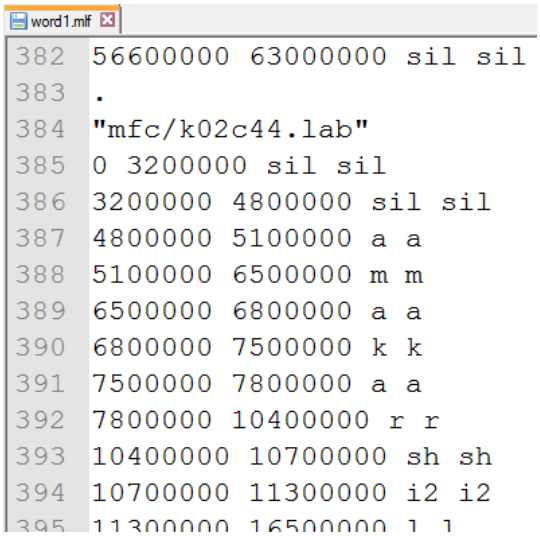
- echo "sil sil" >> dict.list

Ardından 2 eğitim işlemi daha uygulanır:

- ./HERest.exe -A -D -T 1 -C comp.cfg -I phone2.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm07/macros -H hmm07/hmmdefs -M hmm08 phone1.list
- ./HERest.exe -A -D -T 1 -C comp.cfg -I phone2.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm08/macros -H hmm08/hmmdefs -M hmm09 phone1.list

Son işlem olarak HVite aracıyla fonem bazında zaman damgalarını (Resim 3.24) üretme işlemi yapılır:

- HVite.exe -o SM -b sil -C comp.cfg -a -H hmm09/macros -H hmm09/hmmdefs -i word1.mlf -m -t 250.0 -y lab -I word0.mlf -S train.scp -L mfc/ dict.list phone1.list



Resim 3.24. word1.mlf dosyası

Bütün işlemlerin tamamlanmasıyla beraber Türkçe konuşma tanıma sistemlerinde kullanılabilecek bir konuşma veritabanı oluşturulmuştur. 30 kişiye 60'ar cümlenin okutulmasıyla elde edilen 1800 cümleyi içermektedir. Salt ses verilerinin sıkıştırılmamış boyutu 479MB, toplam süresi yaklaşık 4 saat 52 dakikadır.

Konuşmacıların memleketlerine göre dağılımı Çizelge 3.10'da verilmiştir.

Çizelge 3.10. Konuşmacıların memleketlerine göre dağılımı

Memleket	Sayı
Kırşehir	7
Ankara	5
Kayseri	3
Giresun	2
Bursa	1
Osmaniye	1
Sivas	1
Nevşehir	1
Zonguldak	1
Denizli	1
Yozgat	1
Aydın	1
Konya	1
Çorum	1
Trabzon	1
Antalya	1
Mardin	1

Konuşmacıların memleketlerinin bölgelere göre dağılımı Çizelge 3.11'de verilmiştir.

Çizelge 3.11. Konuşmacıların memleketlerinin bölgelere göre dağılımı

Memleket Bölge	Sayı
İç Anadolu	19
Karadeniz	5
Akdeniz	2
Ege	2
Marmara	1
Güneydoğu Anadolu	1

Konuşmacıların yetiştikleri şehirlere göre dağılımı Çizelge 3.12'de verilmiştir.

Çizelge 3.12. Konuşmacıların yetiştikleri şehre göre dağılımı

Yetiştiği Şehir	Sayı
Ankara	8
Kırşehir	6
İstanbul	2
Kayseri	2
Muğla	1
Bursa	1
Adana	1
Nevşehir	1
Zonguldak	1
Denizli	1
Balıkesir	1
İzmir	1
Kocaeli	1
Aydın	1
Antalya	1
Mardin	1

Konuşmacıların yetiştikleri şehirlerin bölgelere göre dağılımı Çizelge 3.13'te verilmiştir.

Çizelge 3.13. Konuşmacıların yetiştikleri şehirlerin bölgelere göre dağılımı

Yetiştiği Bölge	Sayı
İç Anadolu	17
Marmara	5
Ege	4
Akdeniz	2
Karadeniz	1
Güneydoğu Anadolu	1

Konuşmacıların doğum yıllarına göre dağılımı Çizelge 3.14'te verilmiştir.

Çizelge 3.14. Konuşmacıların doğum yıllarına göre dağılımı

Doğum Yılı	Sayı
1993	9
1992	8
1995	5
1994	4
1991	2
1989	1
1990	1

4. SONUÇ VE ÖNERİLER

Bu çalışmada, konuşma tanıma başta olmak üzere, Türkçe dili üzerine yapılacak ses çalışmalarına destek olması amacıyla bir konuşma veritabanı oluşturulmuştur.

Çalışmanın kuramsal temelini Saklı Markov Modeli ve üçlüsesler oluşturmaktadır. Saklı Markov Modelleri, görünür çıktılar üreten, ancak arka planda bu çıktıların üretilmesine neden olan saklı durum geçişlerini içeren durumları modellemek için geliştirilmiştir. Konuşma esnasında, her fonemin kendisinden önceki fonemin söylenişinden etkilenmesi ve kendisinden sonraki fonemin söylenişini etkilemesi nedeniyle üçlüses yapısı tercih edilmiştir.

Bu çalışma için Türkçe dilinin üçlüses istatistiklerinin elde edilmesi için "Sözcük Altı İstatistik" isimli bir yazılım geliştirilmiştir. Geliştirilen bu yazılım .Net Framework 4.0 altyapısını kullanmaktadır ve büyük metinler üzerinde üçlüses istatistik işlemlerinin hızlı bir şekilde yapılabilmesi için optimize edilmiştir. Bu yazılımla elde edilen veriler (üçlüses, kelime ve cümle seviyesinde) tezde sunulmuştur. Bu yazılım kullanılarak, 2 milyon kelimelik bir metinden özgün bir yöntemle seçilen 60 cümle, 30 konuşmacıya okutulmuş ve 1800 ses dosyası elde edilmiştir. Ses dosyaları üzerinde gürültü azaltma işlemi yapılmıştır. Geliştirilen konuşma veritabanı, 1800 wav dosyası ve bunların transkripsiyonlarını içeren lab dosyalarından oluşmaktadır. Geliştirilen bu konuşma veritabanı HTK hedef alınarak geliştirilmesine rağmen, farklı yazılımlarla beraber de kullanılabilir. Bu konuşma veritabanının HTK ile nasıl kullanılacağına dair http://htk.eng.cam.ac.uk/docs/docs.shtml adresinden ulaşılabilen HTKBook isimli başvuru kitabı yardımcı olabilir.

Bu yazılımda üçlüses denge ve zenginliğini koruyabilmek için özgün bir algoritma kullanılmıştır. Kullanılan algoritma tarafından seçilen cümlelerin, 2 milyon kelimelik metindeki üçlüsesleri ne kadar karşıladığının anlaşılabilmesi için karşılaştırmalar yapılmıştır. 2 milyon kelimelik derlemde en sık geçen 40 üçlüses baz alındığında, algoritma tarafından seçilen cümlelerin 37/40 oranında başarılı olduğu görülmüştür. Seçilen cümlelerde j foneminin bulunmadığı tespit edilmiştir. Bu eksikliğin sebepleri, algoritmanın tamamen üçlüses tabanlı çalışması ve seçilen cümle sayısının az olmasıdır. 30

konuşmacının her birine aynı 60 cümlenin okutulması yerine, 1800 cümle seçip her konuşmacıya bunların içinden birbirinden farklı 60'ar tanesinin okutulması daha iyi bir çözüm olabilir.

Türkçe için oluşturulmuş olan konuşma veritabanı sayısı oldukça azdır. Hazır konuşma veritabanları, Türkçe için KT sistemi geliştirmeye çalışan araştırmacılar için son derece önemlidir. Bunların sayı ve çeşitliliğinin artması gerekir. Bir kıyas olması açısından yapılan bir çalışmada [25], Arapça dili için geliştirilmiş 25 konuşma veritabanı ve 29 metin veritabanı tespit edilmiştir. Bu da Türkçe için konuşma veritabanı ile ilgili yapılacak araştırmaların ne kadar önemli olduğunu göstermektedir.

Konuşma veritabanı oluşturabilmek için, öncelikle doğru cümlelerin seçilmesi gerekir. Bunun yapılabilmesi için de büyük metin derlemleri üzerinde kelime altı istatistik çalışmalarının yapılması ve bunların sonucuna göre Türkçe'nin üçlüses yapısını en iyi temsil eden cümlelerin kullanılması gerekir. Şu anda Türkçe konuşma veritabanı geliştirmek isteyen araştırmacıların önünde pek fazla metin derlemi seçeneği yoktur. Bunların sayı ve çeşitliliğinin (metnin derlendiği kaynak bakımından) artması gerekir.

Ayrıca, Türkçe geniş bir coğrafyada konuşulan, lehçe ve ağız bakımından çeşitliliğe sahip bir dil olduğundan, geliştirilen KT sistemleri, tüm lehçe ve ağızlar için aynı performans ile çalışamayacaktır. Bu nedenle, Türkçe'de yer alan lehçe ve ağızların haritası çıkarılmalı ve telaffuz kuralları belirlenmelidir. Bu sayede farklı lehçe ve ağızlara destek veren KT uygulamaları geliştirilebilir.

Böylece yapılan bu çalışma -Türkçe konuşma veritabanı oluşturulmuş olmasından dolayı- önemli bir eksikliği gidermiştir. Bunun yanında bundan sonra konuşma tanıma konusunda yapılacak çalışmalarda bir kaynak niteliğinde olmuştur.

KAYNAKLAR

1. Yalçın, N. (2008). Konuşma Tanıma Teorisi Ve Teknikleri. *Kastamonu Eğitim Dergisi*, 249-266.

2. Aşlıyan, R., Günel, K., Yakhno, T. (2008). Dinamik Zaman Bükmesi Yöntemiyle Hece Tabanlı Konuşma Tanıma Sistemi. *Akademik Bilişim*, 20.

3. Yalçın, N. (2006). *Konuşma Tanıma Teknolojisi Yardımıyla İlköğretim Birinci Sınıf Öğrencilerine İlkokuma Yazma Öğretimi İçin Bir Yazılım Geliştirme*, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 18-19.

4. Nabiyev, V. (2005). *Yapay Zeka*. Ankara: Seçkin Yayıncılık, 710.

5. Salor, Ö., Çiloğlu T., Demirekler M. (2006). ODTÜ Türkçe Mikrofon Konuşması Veritabanı. *IEEE 14. Sinyal İşleme ve İletişim Uygulamaları (SIU)*.

6. Çiloğlu, T., Acar, D., Tokatlı, A., (2004). OrienTel: Türkçe Telefon Konuşması Veritabanı. *IEEE 12. Sinyal İşleme ve İletişim Uygulamaları (SIU)*.

7. Güner, L. (1999). *Türkçe Konuşma ve Konuşmacı Tanımaya Yönelik Veri Tabanı Örnekleminin Oluşturulması*, Yüksek Lisans Tezi, Ankara Üniversitesi Sağlık Bilimleri Enstitüsü, Ankara.

8. Aksoylar, C., Mutluergil, S., O., Erdoğan, H. (2009). Bir Türkçe konuşma tanıma sisteminin anatomisi. *IEEE 17. Sinyal İşleme ve İletişim Uygulamaları (SIU)*.

9. Oflazoğlu, C., Yıldırım, S. (2011). Türkçe Duygusal Konuşma Veritabanı. *IEEE 19. Sinyal İşleme ve İletişim Uygulamaları (SIU)*, 1153-1156.

10. Öcal, K. (2005). *Otomatik Konuşma Tanıma Algoritmalarının Uygulamaları*, Yüksek Lisans Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 3-5.

11. Jelinek, F. (1997). Statistical Methods for Speech Recognition. *MA:MIT*, 15-37.

12. Stratonovich, R.L. (1960). Conditional Markov Processes. *Theory of Probability and its Applications 5*, 156–178.

13. İnternet: Linguistic Data Consortium-Top Ten LDC Corpora. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fcatalog.ldc.upenn.edu%2Ftopten&date=2014-07-02, Son Erişim Tarihi: 2 Temmuz 2014.

14. İnternet: Linguistic Data Consortium-TIMIT Acoustic-Phonetic Continuous Speech Corpus. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fcatalog.ldc.upenn.edu%2FLDC93S1&date=2014-07-02, Son Erişim Tarihi: 2 Temmuz 2014.

15. Ergenç, İ. (2002). *Konuşma Dili ve Türkçe'nin Söyleyiş Sözlüğü*, İstanbul: Multilingual, 39-45.

16. Say, B. (2006). *Türkçe için bir Derlem Geliştirme*. Bilgisayar Destekli Dilbilim Çalıştayı Bildirileri, Ankara: Türk Dil Kurumu Yayınları, No: 868, 81-88.

17. Özge, U., Bilge S. (2004). *Development of a Corpus Workbench for the METU Turkish Corpus*, In Proceedings of 4th Language Resources and Evaluation Conference (LREC 2004), 223-225.

18. İnternet: Google Play'de Android Uygulamaları-Smart Voice Recorder. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fplay.google.com%2Fstore%2Fapps%2Fdetails%3Fid%3Dcom.andrwq.recorder&date=2014-07-02, Son Erişim Tarihi: 2 Temmuz 2014.

19. İnternet: HTK Speech Recognition Toolkit. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fhtk.eng.cam.ac.uk&date=2014-07-02, Son Erişim Tarihi: 2 Temmuz 2014.

20. İnternet: Documentation For HTK. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fhtk.eng.cam.ac.uk%2Fdocs%2Fdocs.shtml&date=2014-07-02, Son Erişim Tarihi: 2 Temmuz 2014.

21. İnternet: Audacity: Windows. URL: http://www.webcitation.org/query?url=http%3A%2F%2Faudacity.sourceforge.net%2Fdownload%2Fwindows&date=2014-07-02, Son Erişim Tarihi: 2 Temmuz 2014.

22. İnternet: Gorman, K. Automatic speech segmentation with HTK. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.ling.upenn.edu%2F%7Ekgorman%2Fspeechseg.html%2F.speechseg.html&date=2014-07-02, Son Erişim Tarihi: 2 Temmuz 2014.

23. İnternet: Download HTK. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fhtk.eng.cam.ac.uk%2Fdownload.shtml&date=2014-07-02, Son Erişim Tarihi: 2 Temmuz 2014.

24. İnternet: Installing HTK on Microsoft Windows. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fhtk.eng.cam.ac.uk%2Fdocs%2Finst-win.shtml&date=2014-07-02, Son Erişim Tarihi: 2 Temmuz 2014.

25. Abushariah, M. A., Ainon, R. N., Zainuddin, R., Alqudah, A. A., Ahmed, M. E., Khalifa, O. O. (2012). Modern standard Arabic speech corpus for implementing and evaluating automatic continuous speech recognition systems, *Journal of The Franklin Institute-engineering and Applied Mathematics*, 349(7), 2215-2242.

EKLER

EK-1. Telaffuz sözlüğünün (Pronunciation Dictionary) hazırlanması

Bu program, cümleler.txt dosyasını işler ve pdict.txt dosyasını oluşturur:

```
string[,] degistir = { { "ç", "ch" }, { "ğ", "g2" }, { "ı", "i2" }, { "ş", "sh" }, { "ö", "oe" }, { "ü", "ue" } };

string txt = File.ReadAllText(@"..\..\..\cümleler.txt");
txt = txt.Replace("\r", "\n");
txt = txt.Replace(".", "");
txt = txt.Replace(",", "");

string[] cumleler = txt.Split("\n".ToCharArray(),
StringSplitOptions.RemoveEmptyEntries);
List<string> kelimeler = new List<string>() { };
List<string> telaffuzlar = new List<string>() { };

string[] gecici;
foreach (string cumle in cumleler)
{
        gecici = cumle.Split(" ".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);
        foreach (string kelime in gecici)
        {
                kelimeler.Add(kelime);
        }
}

kelimeler.Add("silence");
List<string> uniq = kelimeler.Distinct().ToList();
uniq.Sort(StringComparer.OrdinalIgnoreCase);
StreamWriter sw = new StreamWriter(@"..\..\..\pdict.txt", false);

for (int i = 0; i < uniq.Count; i++)
{
        telaffuzlar.Add("");
```

EK-1. (devam) Telaffuz sözlüğünün (Pronunciation Dictionary) hazırlanması

```
        foreach (char harf in uniq[i])
        {
                telaffuzlar[i] += harf.ToString() + " ";
        }
}

for (int i = 0; i < telaffuzlar.Count; i++)
{
        for (int j = 0; j < degistir.GetLength(0); j++)
        {
                telaffuzlar[i] = telaffuzlar[i].Replace(degistir[j, 0], degistir[j, 1]);
        }
}

for (int i = 0; i < uniq.Count; i++)
{
        sw.Write(uniq[i].PadRight(25));
        if (uniq[i] != "silence")
        {
                sw.Write(("[" + uniq[i] + "]").PadRight(25));
                sw.WriteLine(telaffuzlar[i]);
        }
        else
        {
                sw.Write(("[]").PadRight(25));
                sw.WriteLine("sil");
        }
}

sw.Close();
```

EK-2. Kelime listesinin (Word List) oluşturulması

Bu program, cümleler.txt dosyasında yer alan tüm kelimelerin bir listesini oluşturur. Tekrarları eler, kelimeleri alfabetik sıralar (alfabetik sıralama ASCII setine göre yapılmalıdır). Elde edilen sonucu word.list dosyasına kaydeder:

```
string txt = File.ReadAllText(@"..\..\..\cümleler.txt");
txt = txt.Replace("\r", "\n");
txt = txt.Replace(",", "");
txt = txt.Replace(".", "");

string[] cumleler = txt.Split("\n".ToCharArray(),
StringSplitOptions.RemoveEmptyEntries);
List<string> kelimeler_tekil = new List<string>() { };
kelimeler_tekil.Add("silence");
string[] gecici;
foreach (string cumle in cumleler)
{
   gecici = cumle.Split(" ".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);
   foreach (string kelime in gecici)
   {
      if (!kelimeler_tekil.Contains(kelime))
      {
         kelimeler_tekil.Add(kelime);
      }
   }
}

kelimeler_tekil.Sort(StringComparer.OrdinalIgnoreCase);
StreamWriter sw = new StreamWriter(@"..\..\..\word.list", false);
for (int i = 0; i < kelimeler_tekil.Count; i++)
{
   sw.WriteLine(kelimeler_tekil[i]);
} sw.Close();
```

Ek-3. Hmmdefs dosyasının üretilmesi

Bu program, proto ve phone0.list dosyalarını girdi olarak alır, hmm model tanımlamalarını içeren hmmdefs dosyasını üretir:

```
string txt = File.ReadAllText(@"..\..\..\phone0.list");
txt = txt.Replace("\r", "\n");
string[] fonemler = txt.Split("\n".ToCharArray(),
StringSplitOptions.RemoveEmptyEntries);

txt = File.ReadAllText(@"..\..\..\proto");
int start = txt.IndexOf("<BEGINHMM>");
string proto = txt.Substring(start, txt.Length - start);

StreamWriter sw = new StreamWriter(@"..\..\..\hmmdefs");
for (int i = 0; i < fonemler.Length; i++)
{
   sw.WriteLine("~h \"" + fonemler[i] + "\"");
   sw.Write(proto);
}
sw.Close();
```

Ek-4. Klavuz

Bu bölümde, konuşma veritabanının HTK ile eğitimi ve kullanımı öz bir biçimde verilecektir. Verilen komutlar cygwin ortamında, scratch dizini altında çalıştırılmalıdır. HTK ile konuşma tanıma sistemi geliştirilmesi ile ilgili detaylı bilgi almak için HTKBook [20] isimli doküman incelenmelidir.

1. Hazırlık: Gerekli Dosya ve Dizinlerin Oluşturulması

- mkdir hmm00
- mkdir hmm01
- mkdir hmm02
- mkdir hmm03
- mkdir hmm04
- mkdir hmm05
- mkdir hmm06
- mkdir hmm07
- mkdir hmm08
- mkdir hmm09
- mkdir hmm10
- mkdir hmm11
- mkdir hmm12
- mkdir hmm13
- mkdir hmm14
- mkdir hmm15
- mkdir mfc
- ./HDMan -m -w word.list -n phone1.list -l dlog -C myconfig dict.list pdict.txt
- grep -v 'sp' phone1.list > phone0.list
- perl ps/label2mlf data/*.lab > word0.mlf
- ./HLed -l 'data/' -d dict.list -i phone0.mlf mkPhones0.led word0.mlf
- perl ps/copyList > copy.scp

2. Ses Dosyalarından MFCC Öznitelik Vektör Çıkarımı

- ./HCopy -T 1 -C copy.cfg -S copy.scp

Ek-4. (devam) Klavuz

3. Teksesli Eğitimi

- perl ps/compList > train.scp
- ./HCompV -C comp.cfg -f 0.01 -S train.scp -M hmm00 proto
- cat mactmp.txt hmm00/vFloors > hmm00/macros
- perl ps/init phone0.list > hmm00/hmmdefs
- perl ps/phone2label phone0.mlf
- ./HERest -A -D -T 1 -C comp.cfg -I phone0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm00/macros -H hmm00/hmmdefs -M hmm01 phone0.list
- ./HERest -A -D -T 1 -C comp.cfg -I phone0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm01/macros -H hmm01/hmmdefs -M hmm02 phone0.list
- ./HERest -A -D -T 1 -C comp.cfg -I phone0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm02/macros -H hmm02/hmmdefs -M hmm03 phone0.list

4. sp Modelinin Oluşturulması

- perl ps/sil2sp hmm03/hmmdefs > hmm04/hmmdefs
- cp hmm03/macros hmm04/macros
- ./HHEd -H hmm04/macros -H hmm04/hmmdefs -M hmm05 sil.hed phone1.list
- ./HLed -l 'data/' -d dict.list -i phone1.mlf mkPhones1.led word0.mlf
- perl ps/phone2label phone1.mlf

5. sp'li Modellerin Eğitimi

- ./HERest -A -D -T 1 -C comp.cfg -I phone1.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm05/macros -H hmm05/hmmdefs -M hmm06 phone1.list
- ./HERest -A -D -T 1 -C comp.cfg -I phone1.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm06/macros -H hmm06/hmmdefs -M hmm07 phone1.list
- ./HVite -A -D -T 1 -o SWT -b silence -C comp.cfg -a -H hmm07/macros -H hmm07/hmmdefs -i phone2.mlf -m -t 250.0 -y lab -I word0.mlf -S train.scp -L data/ dict.list phone1.list
- ./HERest -A -D -T 1 -C comp.cfg -I phone2.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm07/macros -H hmm07/hmmdefs -M hmm08 phone1.list
-

Ek-4. (devam) Klavuz

- ./HERest -A -D -T 1 -C comp.cfg -I phone2.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm08/macros -H hmm08/hmmdefs -M hmm09 phone1.list

6. Tekses Hizalama İşlemi

- ./HVite -o SM -b silence -C comp.cfg -a -H hmm09/macros -H hmm09/hmmdefs -i word1.mlf -m -t 250.0 -y lab -I word0.mlf -S train.scp -L data/ dict.list phone1.list

7. Üçlüseslere Geçiş

- HLEd -n triphones1 -i wintri.mlf mktri.led phone2.mlf
- perl ps/maketrihed monophones1 triphones1
- HHEd -A -D -T 1 -H hmm9/macros -H hmm9/hmmdefs -M hmm10 mktri.hed monophones1
- HERest -A -D -T 1 -C config -I wintri.mlf -t 250.0 150.0 3000.0 -S train.scp -H hmm10/macros -H hmm10/hmmdefs -M hmm11 triphones1
- HERest -A -D -T 1 -C config -I wintri.mlf -t 250.0 150.0 3000.0 -s stats -S train.scp -H hmm11/macros -H hmm11/hmmdefs -M hmm12 triphones1

8. Bağlı Durumlu Üçlüseslerin Oluşturulması ve Eğitimi

- HDMan -A -D -T 1 -b sp -n fulllist -g global.ded -l flog dict-tri dict.list
- cat fulllist triphones1 >> fulllist1
- perl fixfulllist.pl fulllist1 fulllist
- perl mkclscript.prl TB 350 monophones0 > tree-2.hed
- cat tree-1.hed tree-2.hed > tree-3.hed
- cat tree-3.hed tree-4.hed > tree.hed
- HHEd -A -D -T 1 -H hmm12/macros -H hmm12/hmmdefs -M hmm13 tree.hed triphones1
- HERest -A -D -T 1 -T 1 -C config -I wintri.mlf -s stats -t 250.0 150.0 3000.0 -S train.scp -H hmm13/macros -H hmm13/hmmdefs -M hmm14 tiedlist
- HERest -A -D -T 1 -T 1 -C config -I wintri.mlf -s stats -t 250.0 150.0 3000.0 -S train.scp -H hmm14/macros -H hmm14/hmmdefs -M hmm15 tiedlist

Ek-4. (devam) Klavuz

9. Üçlüses Hizalama İşlemi

- HVite -A -D -T 1 -l '*' -a -b silence -m -C config -H hmm15/macros -H hmm15/hmmdefs -m -t 250.0 150.0 1000.0 -I words.mlf -i aligned.mlf-S train.scp dict.list tiedlist

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, Adı : GÜRLER, Barış Safa
Uyruğu : T.C.
Doğum tarihi ve yeri : 16.04.1986 Eskişehir
Medeni hali : Bekâr
Telefon : 0 (386) 812 29 77
e-mail : brsgrlr@hotmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	G.Ü. / Elektronik-Bilgisayar Eğitimi	Devam Ediyor
Lisans	G.Ü. / Bilgisayar Sistemleri Öğretmenliği	2008
Lise	Cumhuriyet Lisesi	2004

Yabancı Dil

İngilizce