



**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

TÜRKÇE METNİ TÜRK İŞARET DİLİNE DÖNÜŞTÜRME

METE CAN YASAN

YÜKSEK LİSANS TEZİ

2014

TÜRKÇE METNİ TÜRK İŞARET DİLİNE DÖNÜŞTÜRME

TURKISH TEXT TO TURKISH SIGN LANGUAGE TRANSLATION

METE CAN YASAN

Başkent Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin
ELEKTRİK-ELEKTRONİK Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2014

“Türkçe Metni Türk İşaret Diline Dönüştürme” başlıklı bu çalışma, jürimiz tarafından, 22/08/2014 tarihinde, **ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI 'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan : Doç. Dr. Hasan OĞUL

Üye (Danışman) : Doç. Dr. Hamit ERDEM

Üye : Yrd. Doç. Dr. Derya YILMAZ

ONAY

22/08/2014

Prof. Dr. Emin AKATA
Fen Bilimleri Enstitüsü Müdürü

TEŐEKKÜR

Bu tez alıŐmasının her aŐamasında tım bilgi birikimlerini paylaŐan, ilgi ve alakasını hi eksik etmeyen deęerli danıŐmanım Sayın Do. Dr. Hamit ERDEM'e, tezin deney aŐamasında yardımlarından dolayı Yahya Özsoy İŐitme Engelliler Ortaokulu Tırke Öğretmeni Sayın Ergun ERCAN'a ve beni her zaman destekleyen, her zaman yanımda olan aileme ve niŐanlım Kübra BEŐPARMAK'a teŐekkür ederim.

ÖZ

TÜRKÇE METNİ TÜRK İŞARET DİLİNE DÖNÜŞTÜRME

Mete Can YASAN

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Son yıllarda, işitme engelli kişilerle iletişimi kolaylaştırmak için akademik düzeyde çalışmalar yapılmaktadır. Bu alanda kullanılan en önemli iletişim yöntemi işaret dilidir. Görsel ve işitsel medya yoluyla iletişimi hızlandırmak ve kolaylaştırmak amacıyla, işaret dilini yazılı metne, ya da yazılı metni işaret diline dönüştürme üzerine çalışmalar yapılmaktadır. Bu tez çalışmasında, sisteme girilen Türkçe metnin, Türk İşaret Dili'ne (TİD) dönüştürülmesi hedeflenmiştir. Yapılan çalışmada sık kullanılan bilgisayarlı çeviri yöntemleri yerine karma bilgisayarlı çeviri yöntemi ve özgün bir eşleşme algoritması kullanılmıştır. Önerilen yöntemde, Türkçe ve TİD'in yapısına uygun bir algoritmayla sisteme girilen metin işaret dili videolarına dönüştürülmüştür. Önerilen algoritma önce bilgisayar ortamında C# dili kullanılarak, yazılıma dönüştürülmüştür. Çalışmanın ikinci aşamasında, geliştirilen yazılım gömülü ve taşınabilir bir donanım olan FriendlyARM Mini2440 geliştirme kartına aktarılmıştır. Geliştirilen algoritmanın başarısı duyma engelliler üzerinde test edilmiş ve sonuçlar yorumlanmıştır.

ANAHTAR SÖZCÜKLER: Türk İşaret Dili, Metin Dosyasından İşaret Diline Dönüştürme, Doğal Dil İşleme, Bilgisayarlı Çeviri, Kural Tabanlı Bilgisayarlı Çeviri, Örnek Tabanlı Bilgisayarlı Çeviri

Danışman: Doç. Dr. Hamit ERDEM, Başkent Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü.

ABSTRACT

TURKISH TEXT TO TURKISH SIGN LANGUAGE TRANSLATION

Mete Can YASAN

Başkent University Institute of Science and Technology

Department of Electrical and Electronics Engineering

In recent years, academic researches have been done to simplify communication with hearing impaired people. Sign language is the most important communication method that is used to achieve this goal. Case studies have been done about sign language to text, or text to sign language conversion to speed up and simplify the communication with using visual and aural media sources. In this research, it is aimed to convert the Turkish text input to Turkish Sign Language (TİD). In this study, a hybrid machine translation method is used instead of a popular machine translation method and a unique matching algorithm was used. In the proposed study, the text input is converted to sign language videos with using an algorithm that is compatible with the architecture of Turkish and TİD. Firstly, the proposed algorithm has been converted into software with using C# in computer environment. In the second part of the study, the developed software has been implemented into FriendlyARM Mini2440, which is an embedded and a portable evaluation board. The success of the developed algorithm has been tested with the help of hearing impaired people and the results have been discussed.

KEY WORDS: Turkish Sign Language, Text to Sign Language Conversion, Natural Language Processing, Machine Translation, Rule Base Machine Translation, Example Based Machine Translation

Advisor: Doç. Dr. Hamit ERDEM, Başkent University, The Department of Electrical and Electronics Engineering

İÇİNDEKİLER

Sayfa No:

ÖZ	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİLLER LİSTESİ	v
ÇİZELGELER LİSTESİ	vi
SİMGELER VE KISALTMALAR LİSTESİ	vii
1. GİRİŞ	1
2. TÜRK İŞARET DİLİ	6
2.1. TİD Kullanıcıları.....	6
2.2. TİD Tarihi	7
2.3. TİD Özellikleri.....	8
2.3.1. Genel özellikler	8
2.3.2. Türkçe ile farklılıklar	10
2.3.3. Dilbilimsel özellikler.....	10
3. BİLGİSAYARLI ÇEVİRME TEKNİKLERİ	15
3.1. Bilgisayarlı Çevirme	15
3.2. Bilgisayarlı Çevirme’de Yaygın Kullanılan Yöntemler.....	16
3.2.1. İstatistiksel bilgisayarlı çevirme	16
3.2.2. Kural tabanlı bilgisayarlı çevirme	17
3.2.3. Örnek tabanlı bilgisayarlı çevirme	20
4. YAZILI METİNDEN TİD’e DÖNÜŞTÜREN SİSTEMİN YAPISI	23
4.1. Kelime Ayırıcı	25
4.2. Kelime Analizi.....	26
4.2.1. Sonek bulma	27
4.2.2. Eşleştirme algoritması.....	28
4.3. Görsel Veri	35
5. UYGULAMALAR	38

6. TESTLER VE SONUÇLAR.....	44
7. SONUÇ ve ÖNERİLER.....	51
KAYNAKLAR.....	53

ŞEKİLLER LİSTESİ

Sayfa No:

Şekil 2.1	Türkiye'de İşitme Engelliler Nüfus Oranı Dağılımı	6
Şekil 2.2	TİD Alfabetesi	9
Şekil 2.3	TİD'de "değil"	12
Şekil 2.4	Soru parçacığı	13
Şekil 2.5	TİD'de Peribacası	14
Şekil 3.1	Bilgisayarlı Çevirme Piramidi	15
Şekil 3.2	Örnek bir cümlenin örnek bir KTBC senaryosunda ayrıştırılması	19
Şekil 3.3	Örnek bir cümlenin örnek bir ÖTBC senaryosunda eşleştirilmesi.....	21
Şekil 4.1	Sistem Blok Diyagramı	24
Şekil 4.2	Akış diyagramı.....	34
Şekil 4.3	Kelime işleme algoritmasının sözde kodu	35
Şekil 4.4	Örnek resim.....	36
Şekil 5.1	Program arayüzü	38
Şekil 5.2	Programda kelime işlenmesi	39
Şekil 5.3	Programda kelimelerin TİD karşılıklarının gösterilmesi.....	39
Şekil 5.4	Kelime dizin örneği	40
Şekil 5.5	Kullanılan donanımın devre yapısı	41
Şekil 5.6	Donanım üzerinde çakar bellek ve giriş metnin gösterilmesi	42
Şekil 5.7	Çalışan donanım üzerinde TİD görüntülerinin gösterimi.....	43
Şekil 6.1	Teste katılan öğrencilerin cümleleri anlama oranı	47
Şekil 6.2	Test sonuçları	48
Şekil 6.3	Testte kullanılan cümlelerin anlaşılma oranı.....	49

ÇİZELGELER LİSTESİ

	<u>Sayfa No:</u>
Çizelge 4.1 Kelime Ayırıştırıcı Örneği	25
Çizelge 4.2 Kelime etiketleme örneği	27
Çizelge 4.3 Sonek bulma örneği	28
Çizelge 4.4 DP Uzaklık Denklemi Parametreleri	29
Çizelge 4.5 Uzaklık Denklemi Parametreleri	30
Çizelge 4.6 Örnek cümle analizi.....	33
Çizelge 5.1 Kullanılan geliştirme kartının donanımsal özellikleri	41
Çizelge 6.1 Çalışmada kullanılan metinler	45
Çizelge 6.2 Testte kullanılan cümlelerin anlaşılma dağılımı (1: anladı, 0: anlamadı)	46

SİMGELER VE KISALTMALAR LİSTESİ

BÇ	Bilgisayarlı Çevirme
DDİ	Doğal Dil İşleme
TİD	Türk İşaret Dili
İBÇ	İstatistiksel Bilgisayarlı Çevirme
ÖTBÇ	Örnek Tabanlı Bilgisayarlı Çevirme
KTBÇ	Kural Tabanlı Bilgisayarlı Çevirme
DP	Dinamik Programlama

1. GİRİŞ

İletişimin insan hayatındaki rolü oldukça büyüktür. Fakat kullanılan diller farklı olduğunda iletişim kurmak zorlaşmaktadır. Özellikle de konuşulan diller ve işaret dili arasındaki farklılık gözetsilirse, bu zorluğun en yüksek derecede olduğu düşünölebilir. İşaret dili görsel bir dildir ve işitme engelli kişiler ile iletişim işaret dili ile sağlanır. Bir konuşulan dili diğetine dönüştürmeye göre, konuşulan bir dili işaret diline dönüştürmek daha büyük zorluklar getirmektedir.

İşitme engelli kişiler çevre ile iletişim kurmakta, bilgi almakta ve sosyal iletişim gibi konularda zorluk çekmektedirler. İşitme engelliler, özellikle de bir dil öğrenmeden işitme engeli olanlar, duyabilen insanlarla iletişim kurarken veya yazılı metinleri anlarken güçlük çekebilirler. İşitme engelli kişiler konuştuklarını duyamazlar. Bu geri besleme mekanizmasının eksikliğinden dolayı da yazılı metinleri anlamakta güçlük çekerler. Buna rağmen, yazılı metin üzerinden haberleşme yine de işitme engelliler ve duyabilen insanlar arasında kullanılan bir yöntemdir. Fakat yürürken, uzaktan ya da hızlı haberleşme gibi durumlar söz konusu olduğunda yazarak haberleşme yetersiz kalır. Dolayısıyla okumaya dayalı haberleşme yöntemleri işitme engelli insanlar ve duyabilen insanlar arasındaki iletişimi sağlamada yetersiz kalmaktadır. Bu tip problemlerin üstesinden gelebilmek için işaret dilleri kullanılır. İşaret dili, konuşulan dillerden farklı ve birçok insanın düşüncesinden farklı olarak, kendi grameri ve yapısal özellikleri olan, başlı başına bir dildir. İşaret dili evrensel bir dil değildir. Örneğin Amerikan İşaret Dili ve Türk İşaret Dili (TİD) farklı kelimelere ve farklı alfabelere sahiptirler. Türkiye'deki işitme engelliler Türkiye'nin farklı bölgelerinden işitme engellilerle TİD kullanarak rahatça anlaşabildiklerini belirtirken, yurtdışından gelen yabancı işitme engellilerle iletişim kurmakta güçlük çektiklerini belirtmişlerdir [1]. Bu da TİD'in kendi grameri olan başlı başına bir dil olduğunu göstermektedir.

TİD, Türkiye'de işitme engelli topluluklarda yaygın olarak kullanılan işaret dilidir. Her ne kadar Türkiye'deki işitme engelli sayısı hakkında kesin bir bilgi olmasa da, bu sayı yaklaşık olarak 2,5 milyon civarındadır [2]. Dünyanın çoğu yerinde olduğu gibi, Türkiye'de de işitme engelliler toplumda azınlıkta kalmaktadır. Bu da, işaret dilinin bilinirliğinin yeterli seviyede olmamasına yol açar. İşaret dilinin duyabilen

insanlar arasında yeterli seviyede bilinir olmaması da, işitme engellilerle duyabilen insanlar arasında iletişim zorluklarının yaşanmasına yol açar. Her ne kadar tercümanlar aracılığı ile bu zorlukların üstesinden gelinebilse de, bu yöntem pahalı ve yetersiz kalmaktadır. Bu da, otomatik olarak çevrim yapabilecek sistemlere yönelik ihtiyacı bir nebze daha artırmaktadır.

İşitme engelliler ve duyabilen insanlar arasındaki iletişim güçlüğüne hafifletmek amacıyla teknolojiden yararlanılarak çalışmalar yapılmaktadır. Bu çalışmalar genel olarak bilgisayarlı çeviri (BÇ) adı altında geçmektedir. BÇ genel olarak iki farklı dilin birbirine otomatik çevrilmesi konusudur. 60 yılı aşkın süredir, konuşulan diller arası bilgisayarlı çeviri yöntemleri kullanılmaktadır [3]. Özellikle son 20 yılda konuşulan dillerden işaret dillerine çevrim üzerine yapılan çalışmalar hız kazanmıştır. Bu çalışmalardan bir kısmı konuşmayı işaret diline çevirmeyi hedeflerken, bir kısmı da yazılı metni işaret diline çevirmeyi hedefler. Bu çevrim farklı doğal dil işleme (DDİ) metotları kullanılarak yapılabilir. Veale ve Conway tarafından 1998'de tasarlanan Zardos isimli sistem konuşma ve yazılı metnin işaret diline çevrilmesi üzerine yapılmış ilk kural tabanlı çalışmalardan biridir [4]. Sisteme girilen İngilizce metin parçalara ayrılır ve söz dizimsel olarak analiz edilir. Analiz sonrasında kural tabanlı bir yapı ile işaret diline dönüşüm sağlanmıştır. 2002'de Marshall ve Sáfár yazılı metinden İngiliz İşaret Dili'ne çevrim yapabilen bir sistem tasarlamışlardır [5]. Çalışmalarında sisteme giren İngilizce metni özel bir program modülü kullanarak (CMU link parser) parçalamışlar ve giriş metnini bu şekilde analiz etmişlerdir. Analiz edilen giriş metni, sisteme entegre edilen hedef ve kaynak diller arasındaki gramer çevrim kuralları kullanılarak çevrilmiştir.

Kural tabanlı çalışmalar olduğu kadar, kural tabanlı olmayan çalışmalar da vardır. Örneğin Bungereoth ve Ney, çalışmalarında Almanca ve Alman İşaret Dili arasında istatistiksel bilgisayarlı çevirme metodu (İBÇ) ile otomatik çeviri yapabilen bir sistem geliştirmiştir [6]. Bazı araştırmacılar da bir diğer farklı metot olan örnek tabanlı bilgisayarlı çevirme (ÖTBÇ) yöntemini kullanarak ilerlemişlerdir. Örneğin Boulares ve Jemni, geliştirdikleri sistemde ÖTBÇ metodunu kullanarak yazılı metni Amerikan İşaret Dili'ne çevirme üzerine çalışmışlardır [7]. Ama son dönemde yapılan çalışmalar göstermektedir ki, tek bir metodu kullanmaktansa, birden fazla metodu harmanlamak daha iyi sonuçlar verebilmektedir [8].

Amerikan İşaret Dili [7, 9] başta olmak üzere, İspanyol İşaret Dili [10], Arap İşaret Dili [11], Yunan İşaret Dili [12] ve Hint İşaret Dili [13] gibi birçok farklı işaret dili üzerine literatürde BÇ çalışması bulmak mümkündür. Fakat ne yazık ki, TİD üzerine yapılan çalışmalar oldukça sınırlıdır. Koç Üniversitesi TİD üzerine kapsamlı bir araştırma yapmış, TİD üzerine genel bilgi veren ve TİD kelime listesini içeren bir internet sitesi geliştirmişlerdir [2].

Boğaziçi Üniversitesi de Türkçe'den TİD'e çeviren bir sözlük geliştirmiştir [14]. Sözlük, veri tabanında saklı olan kelime ve cümlelerin seçilmesiyle ilgili kayıtlı videoyu eşleştirip ekranda gösterir. Bu özelliğiyle sistem oldukça başarılı da olsa, yine de tam bir çevirici niteliği taşımamaktadır.

Cemil Öz ve arkadaşları tarafından yapılan çalışmada İBÇ yöntemi kullanarak konuşma veya yazılı metni TİD'e çeviren bir sistem tasarlanmıştır [15]. İBÇ yönteminin bir parçası olarak Saklı Markov Modeli kullanılmıştır. Bu modelin sisteme aktarılması için hazır bir yazılım kullanılmıştır. Çıkişta TİD animasyonu gösterilmektedir. Çalışmanın donanıma aktarılması hedeflenmediği için hazır yazılım kullanılmasında sakınca görülmemiştir. Yapılan çalışma 40 kelime için test edilmiştir. Cümlenin bütününün anlamsal yeterliliğın kaybedilmeden çevrilmesi konusundaki başarısı belirtilmemiştir.

Önceki çalışmalar [2, 14, 15] sadece bilgisayar tabanlı olarak tasarlanmıştır ve çevirici olarak cümle bazında anlamlı çevirme ihtiyaçlarını karşılayabilecek düzeyde değildir. Bu tez çalışmasının taşınabilir bir platformda da çalışabilmesi ve giriş metninin cümle bazında anlamını kaybetmeden çevrilmesi hedeflenmiştir. Bu amaçla, önerilen algoritmanın hedef ve kaynak dillerin dilbilimsel özelliklerine uygun olmasına dikkat edilmiştir.

Bu tezde önerilen sistemde Türkçe'nin ve TİD'in dilbilimsel özellikleri göz önünde bulundurularak, uygun olabilecek BÇ yöntemi kullanılmıştır. Kural Tabanlı Bilgisayarlı Çevirme (KTBC) kuralların sisteme düzgün aktarılması durumunda oldukça verimli bir sistem olmasına rağmen, giriş metninin oldukça iyi bir şekilde morfolojik analizinin yapılması gereklidir. ÖTBC yönteminde cümle ve cümle kalıpları ile çalışmak sistemin esnekliğı açısından uygun olmayabilir. Çünkü bu durumda sistem sadece tanımlı cümle kalıpları üzerinde doğru eşleştirme

yapabilirken, tanımlı olmayan cümle kalıpları üzerinde başarısız olabilir. Ayrıca sistemin genişletilmesi kolay olmayabilir. İBÇ yönteminde ise sistemin verimli çalışabilmesi için oldukça geniş veri tabanına ihtiyaç duyulur. TİD'deki kelime sayısının diğer işaret dillerine göre oldukça az olduğu düşünülürse bu yöntem TİD için yeteri kadar verimli olmayabilir. Geliştirilen algoritmada örnek ve kural tabanlı yöntemler harmanlanarak ve özgün bir eşleşme algoritması eklenerek doğru eşleştirilen videonun gösterilmesi hedeflenmiştir. BÇ algoritması tasarlanırken, Türkçe'nin sondan eklemeli bir dil olması üzerinde durulmuştur. Kelime analizi bu kavramdan yola çıkarak yapılmıştır. Eğer ilgili kelimenin karşılığının veri tabanında olmadığı ve doğru eşleşme yapılamayacağı belli olursa, sistem bu kelimeleri parmakla heceleyerek çevirecek şekilde tasarlanmıştır. Bu yöntem, kural tabanlı ve örnek tabanlı yöntemlerden esinlenilerek karma bir yöntem şeklinde uygulanmıştır. Kelime analizinde kullanılan eşleşme algoritması, özellikle kelimeler arasındaki farklılığın tespit edilmesi açısından diğer çalışmalardan farklıdır. Bu kısımda örnek tabanlı yöntemlerden esinlenilerek farklı bir yöntem geliştirilmiştir.

Yazılı metinden işaret diline eşleştirme yapıldıktan sonra, işaret dilinin gösterimi için video görüntüleri [11] ya da animasyonlar [5, 7] kullanmak mümkündür. Bu çalışmada işaret dili gösteriminde video parçaları kullanılmıştır. Video parçalarının oluşturulmasında, TİD eğitimi için yazılan kaynak kitap [17] ile beraber kullanılan eğitim videolarından yararlanılmıştır.

Yapılan çalışmada Visual Studio ortamında C# dili ile önce bilgisayar tabanlı yazılım geliştirilmiştir. Geliştirilen yazılım daha sonra donanıma uygun şekilde değiştirilerek donanıma aktarılmıştır. Çalışmada FriendlyARM Mini2440 geliştirme kiti [16] donanım olarak kullanılmıştır. Kit üzerinde ARM mimarisinde güçlü bir işlemci, SD kart soketi, USB arayüzü ve LCD ekran gibi gerekli donanımsal özellikler bulunmaktadır. Sistemde görüntü oynatma, harici bellekten veri okuma gibi donanımsal istekleri pratik bir şekilde gerçekleyebilmek için işletim sistemi kullanılmıştır. İşletim sistemi olarak WinCE seçilmiştir. Visual Studio ile C# yazılım dili kullanılarak kod geliştirilmiştir.

Önerilen çalışmanın amacı, kullanıcı tarafından tanımlanan yazılı metni TİD'e dönüştürebilecek bir sistem tasarlamak ve bu sistemi taşınabilir bir donanım üzerinde gerçekleştirmektir. Yapılan çalışmada giriş metni kelimelere ayrılmıştır.

Tespit edilen kelimeler son eklerine ve köklerine bakılarak ilgili TİD videoları ile eşleştirilmiştir. Kelime analizi ve kök karşılaştırmak için kullanılan yöntemler özgün olma niteliği taşımaktadır. Sistemde bulunmayan kelimeler ve özel isimler parmakla hecelenilerek çevrilirler. Tasarlanılan sistemin halka açık alanlarda kullanılması ve TİD eğitimi amaçlı kullanılarak fayda sağlanması hedeflenmektedir.

Bölüm 2’de TİD kavramına değinilmiş, TİD’in geçmişi, TİD kullanıcıları ve dilbilimsel özellikleri ile TİD’in Türkçe ile olan farklılıklarına yer verilmiştir.

Bölüm 3’te BÇ kavramı üzerine bilgi verilmiştir. Literatürde sıkça kullanılan BÇ teknikleri üzerine bilgi verilmiş, bu tekniklerin olumlu ve olumsuz yönleri üzerinde durulmuştur.

Bölüm 4’te tasarımı yapılan sistem üzerine bilgi verilmiştir. Sistem tasarımında kullanılan algoritma, sistemde kullanılan alt birimler, çalışma prensibi ve kullanılan çevrim yöntemi üzerine bilgi verilmiştir.

Bölüm 5’te tasarımın gerçekleşmesi amacıyla kullanılan yazılım ve donanımlar hakkında bilgi verilmiştir. Ayrıca programın donanımsal olarak çalışma mantığı üzerine de bilgi verilmiştir.

Bölüm 6’da sistemin başarısının belirlenmesi için yapılan testler üzerine bilgi verilmiştir. Kullanılan yöntemler üzerinde durulmuş ve sonuçlar yorumlanmıştır.

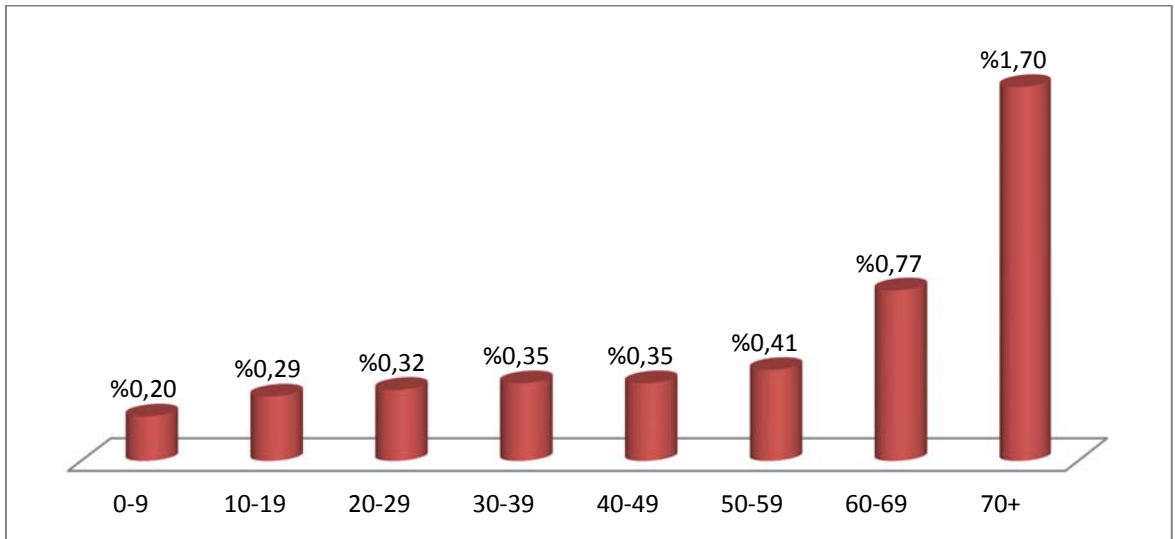
Bölüm 7’de önerilen yöntemin performansı üzerinde durulmuş, yorum ve öneriler sunulmuştur.

2. TÜRK İŞARET DİLİ

Türk İşaret Dili (TİD), Türkiye’de kullanılan işaret dilidir. Bazı bölgesel farklılıklar bulunmasına rağmen, TİD’in tüm Türkiye’de kullanıldığı bilinmektedir. TİD’in farklı bir işaret dilinden türetildiği ya da tarihte farklı bir işaret dilinin etkisinde kaldığı yönünde düşünmek için yeterli kanıt yoktur. Fakat TİD, İngiliz İşaret Dili ile benzerlikler göstermektedir. TİD tarihi, Osmanlı dönemine kadar uzanır. Fakat Milli Eğitim Bakanlığı’nın 1995’te yayınladığı görsel bir kılavuz dışında yazılı bir materyal, sözlük ya da arşiv bulunmamaktadır [2]. TİD tarihinin oldukça eski olduğu bilinmektedir. Buna rağmen TİD için yapılan araştırmalar çok eskiye dayanmamaktadır. TİD’i dilbilimsel olarak ele alan ilk araştırmalar Ulrike Zeshan ve Hasan Dikyuva tarafından yapılmıştır [1, 17]. Dr. Aslı Özyürek ve proje ekibi ise 750 kelimelik bir söz listesini internet ortamında paylaşarak sözlük niteliğinde bir çalışmayı araştırmacılara açmışlardır [2]. Bu bölümde TİD kullanıcıları, TİD tarihçesi ve TİD özelliklerine değinilecektir.

2.1. TİD Kullanıcıları

Türkiye’de ne kadar işitme engelli olduğuna dair kesin bir sayı bulunmamaktadır. Birleşmiş Milletler raporuna göre bu sayı 2,5 milyondur. Şekil 2.1’de, Türkiye İstatistik Kurumu verilerine göre, işaret engelli nüfus yüzdesinin yaş ile arttığı da görülmektedir.



Şekil 2.1 Türkiye’de İşitme Engelliler Nüfus Oranı Dağılımı [18]

Türkiye, işaret dili eğitimi olarak dünya ülkelerinden bir hayli geridedir. TİD eğitimi veren yeterli sayıda okul bulunmaması ve yeterli eğitim materyallerinin bulunmaması TİD öğrenme ve kullanma oranının azalmasına yol açmaktadır. Her şeye rağmen, Türkiye’de işaret engelli cemiyetlerinin sayısı gün geçtikçe artmaktadır. Bu cemiyet ve dernekler Türkiye İşitme Engelliler Milli Federasyonu çatısı altındadır. Bu cemiyet ve dernekler, yurt genelinde spor ve kültürel alanda aktiviteler düzenlemektedir. İşitme engelli dernekleri genelde sosyalleşme amacıyla işitme engellilere hizmet etmektedir. TRT2 başta olmak üzere, birçok ulusal televizyon kanalında da işaret dilinde yapılan yayınlara rastlamak mümkündür.

2.2. TİD Tarihi

TİD, diğer işaret dillerinden birçoğundan daha eski bir işaret dilidir. TİD’in Osmanlı saraylarında yaygın olarak kullanıldığı bilinmektedir [19]. “Dilsiz” olarak bilinen işitme engellilerin Fatih Sultan Mehmed döneminden beri sarayda görev yaptıkları bilinmektedir. Bu kişiler özellikle güvenlik ve sultanın eğlendirilmesi gibi konularda görev yaptıkları bilinmektedir. İşitme engelliler 19. Yüzyıla kadar sadece saraylarda istihdam edilirken, 19. Yüzyıl ve sonrasında gizli meselelerin görüşüldüğü toplantılarda da görev almaya başlamışlardır. İşitme engelli kişilerin sarayda padişahla, saray erkânıyla ve kendi aralarında özel işaretlerle anlaştıkları da bilinmektedir.

Özellikle 17. Yüzyıl’da işaret dilinin sarayda kullanımı oldukça yaygınlaşmıştır. Hatta işaret dili sadece işitme engelliler tarafından değil, duyabilen insanlar tarafından da kullanılmaya başlanmıştır [19]. Osmanlı’da kullanılan bu işaret dili, batıdaki işaret dillerinden bağımsız olarak gelişen, özgün bir dil olarak bilinmektedir [2].

Osmanlı Devleti’nde ilk işaret dili okulu padişah 2. Abdülhamit tarafından 1902’de açılmıştır. Bu okulda Osmanlı İşaret Dili’nin sözel dil ile beraber kullanıldığı bilinmektedir. Her ne kadar o yıllarda kullanılan Osmanlı İşaret Dili’nin günümüzde

kullanılan Türk İşaret Dili'nin alt yapısını oluşturduğu düşünülse de, Osmanlı İşaret Dili'nde kullanılan alfabe ile TİD alfabesi oldukça farklıdır.

1953'te Milli Eğitim Bakanlığı tarafından çıkarılan bir kanunla işaret dilinin okullarda kullanımı yasaklanmıştır. Böylece işaret dilinin yerine sözel eğitim verilerek sözel eğitimin işitme engelliler arasında yaygınlaştırılması ve erken yaştakilerin konuşmasının mümkün olabileceği düşünülmüştür. Fakat bu metodun yanlış olduğunun farkına daha sonra varılmıştır. Bu yasak 2005'te kaldırılmıştır. Halen TİD'nin gelişmesini ve yaygınlaştırılmasını sağlamak amacıyla çalıştaylar düzenlenmektedir. TİD üzerine yapılan dilbilimsel araştırmalar çok eski tarihlere dayanmamaktadır. İlk TİD sözlüğü 1993'te Ankara'da çıkarılmıştır. Şu an mevcut olan dilbilimsel çalışmalar yeterli değildir ve mevcut TİD sözlükleri tam ve düzgün olarak çalışmamaktadır. TİD konusunda yapılan çalışmaların giderek artması beklenmektedir.

2.3. TİD Özellikleri

TİD, kendi dilbilimsel özellikleri olan ve işitme engelliler tarafından birinci dil olarak kullanılan dil olma özelliğini taşır. TİD'in kendine özgü ifade araçları olduğu ve kullanıcıları tarafından özgün bir şekilde oluşturulduğu bilinmektedir. Bu özellikleriyle TİD, sanılanın aksine, Türkçe'nin işaretlerle taklit edilmesinden öte, farklı bir dil olarak değerlendirilmelidir. TİD, Türkçe'den türetildiği için tabii ki Türkçe ile benzerlikler göstermektedir. Fakat cümlelerdeki karmaşıklık arttıkça bu benzerliklerin azaldığı görülmektedir.

2.3.1. Genel özellikler

TİD, genel olarak kelimelerle ifade edilen bir dildir. TİD'de 750 civarında kelime tespit edilmiştir [2]. Bu kelimeler değişik kategorilerle sınıflandırılmıştır. Bu kategoriler; alfabe, sayılar, zamanla ilgili kavramlar, görsel kavramlar, hayvanlar, meslekler, yer isimleri, zamirler ve anlatımlar olarak belirtilmiştir. Genelde TİD belirlenmiş olan bu kelimeler üzerinden ifade edilmektedir. Fakat kelimelerin yeterli olmadığı durumlarda ya da özel isimlerin geçtiği yerlerde harf işaretleri ile heceleme yapılmaktadır. Türkçe'de yer alan harflerin hepsinin TİD'de bir karşılığı vardır. Tüm harfler tek tek el hareketleri ile ifade edilebilmektedir. Bu da TİD'e el

hareketlerinden oluşan bir alfabe kazandırmaktadır. Şekil 2.2'de TİD Alfabeti gösterilmiştir.



Şekil 2.2 TİD Alfabeti [1]

2.3.2. Türkçe ile farklılıklar

TİD, temel dilbilimsel özellikleri açısından Türkçe'den ayrılmaktadır. Fakat söz dizimi ve anlamsal açıdan bakıldığında TİD ile Türkçe birçok açıdan benzerlikler göstermektedir. Asıl farklılıklar çoğullaştırma, olumsuzlaştırma, zaman gibi dilbilimsel özellikler açısından bakıldığında görülmektedir. Bu konudaki detaylar bir sonraki bölümde anlatılacaktır.

Türkçe, sondan eklemeli bir dil yapısına sahiptir. Yani, sözcükler sonlarına eklenen ekler ile türetilirler veya farklı anlamsal nitelikler kazanabilirler. Fakat TİD kelime tabanlı bir dildir. Yani cümle içindeki her anlam ayrı bir kelime ile ifade edilir. Bu farklılığa bir sonraki bölümde detaylıca değinilecektir.

Bir diğer farklılık ise, Türkçe'de tamamlanmışlık görünüşü için özel bir kelime bulunmazken, TİD'de cümlenin sonuna "bitti" ya da "tamam" getirilerek tamamlanmışlık görünüşü kazandırılabilir. Bu gibi özellikleri ile TİD, Türkçe'den farklı bir dil olarak değerlendirilmelidir.

2.3.3. Dilbilimsel özellikler

TİD ve konuşulan Türkçe dilbilimsel olarak birbirlerine göre farklılık gösterebilir. Yaygın kanıya göre, TİD konuşulan Türkçe'nin işaretlerle taklit edilmesinin bir sonucu olacak doğmuştur. Fakat bu kanı doğru değildir. Zira TİD kendine özgü dilbilimsel özellikleri bulunan farklı bir dildir. Fakat her ne kadar TİD farklı bir dil de olsa, TİD kullanıcıları izole bir ortamda olmayıp, Türkçe konuşabilen kişilerle bir arada yaşamakta ve hatta eğitimlerini normal öğrenciler ile aynı metotlarla almaktadırlar. Bu gibi nedenler, TİD ile konuşulan Türkçe arasında yüksek bir etkileşim olmasına yol açmaktadır.

TİD ve konuşulan Türkçe arasında gözlemlenen önemli farklılıklardan birisi sondan ekleme özelliğidir. Türkçe sondan eklemeli bir dildir. Fakat TİD'nin dil bilgisi ile ilgili özelliklerine baktığımızda sondan eklemeli olma özelliğini göremeyiz. Türkçe'de, özellikle yüklemelerde, cümlenin anlamına katkıda bulunan birçok cümle ögesi ek şeklinde kodlanabilir. Mesela Türkçe'de sık olarak zamirlerin yüklem içinde ek olarak kodlandığı görülür. Fakat TİD'de zamirler yüklem üzerine kodlanamaz. Örnek vermek gerekirse, "çalışıyorum" cümlesi içerisinde zamirin yükleme ek

olarak kodlandığı görülür. Fakat aynı cümlelerin TİD karşılığı “ben + çalışmak” ya da “ben + çalışmak + ben” şeklindedir. Örnekte de görüldüğü gibi, TİD’de zamirler yükleme ek olarak kodlanmak yerine ayrı bir kelime olarak cümleye eklenmiştir. Örnekte de dikkat çeken bir diğer konu, zamirin cümle içinde sona gelmesi ve hatta tekrarlanabilmesidir. Zamirin sona gelmesi veya tekrarlanabilmesi tamamen kullanıcıya bağlıdır. Zamirin vurgulanması gerektiği durumlarda tekrarlama yapmak daha uygun olabilir.

TİD ve konuşulan Türkçe ile arasındaki farklılıklardan biri de zaman kavramının cümle içinde belirtilme şeklidir. Türkçe’de zaman kavramı genellikle yükleme ek olarak kodlanır. Fakat TİD’de bu durum farklılık gösterir. Dilbilimci Zeshan, çalışmalarında işaret dillerinde dilbilimsel zamana pek rastlanılmadığını belirtir [1]. Cümledeki zaman kavramının “önce”, “sonra” gibi zaman anlamı içeren sözcüklerin yanı sıra “tamam” ve “bitti” gibi tamamlanmışlık anlamı içeren sözcükler tarafından katıldığı görülür. Örnek vermek gerekirse, “geliyorum” ile “geleceğim” cümlelerinin TİD çevrimleri aynıdır. Fakat cümle içine eklenen zaman anlamı içeren sözcük ve ya kelime öbekleri ile zaman bilgisi cümleye eklenebilir.

TİD’de olumsuzluk anlamı genelde başı yukarı doğru kaldırarak, kaşları yukarı doğru kalkık vaziyette, tek eli ya da her iki eli havaya kaldırarak verilir. Bu hareket aynı zamanda “değil” anlamına da gelmektedir. Şekil 2.3’te, TİD’de “değil” anlamına gelen hareket gösterilmiştir. Değil anlamına gelen bir diğer hareket de başı iki yana doğru sallamaktır. Fakat yaygın olarak ilk seçenek kullanılır. Diğer anlamlarda da olduğu gibi, olumsuzluk anlamının da Türkçe’de cümle içinde kodlanmış olarak geldiği görülür. Olumsuzluk içeren sıfatlarda durum Türkçe ile TİD arasında benzerdir. Her ikisinde de sıfatın sonuna “değil” eklendiği görülür. Fakat, olumsuzluk anlamı içeren yüklemelerde durum farklıdır. Türkçe’de fiili olumsuzlaştırmak için fiile –me, -ma gibi ekler gelirken, TİD’de “fill + değil” şeklinde olumsuzluk anlamı katılır. Örnek olarak, “gitmemek” sözcüğü TİD’de “gitmek + değil” şeklinde çevrilecektir.



Şekil 2.3 TİD’de “değil”

TİD’de soru sorma yöntemleri de dilbilimsel olarak Türkçe ‘de olduğundan farklıdır. Soru cümlelerini kendi içinde Evet/Hayır soruları ve soru kelimeleri ile sorulan sorular olarak ikiye ayırabiliriz. Türkçe’deki Evet/Hayır soru tiplerine bakıldığında, bir cümlenin soru anlamı kazanmasına yol açan etmenin cümlenin sonuna eklenen soru parçacığı olduğu görülmektedir. Yani cümlenin sonunda yer alan –mi, -misin gibi soru parçacıkları sıradan bir cümleyi soru cümlesine çevirmektedir. Bu tip soru cümlelerini TİD’e çevirmek Zeshan’ın araştırmasına göre iki farklı yoldan mümkün olabilir [1]. İlk yöntem cümleyi olduğu gibi çevirip, sonuna Türkçe’deki soru parçacığına denk gelen bir işaret eklemektir. Bu işaret Şekil 2.4’te gösterilmiştir. Bu yöntem Türkçe’deki yöntem ile oldukça benzerdir. Bir diğer yöntem ise, cümleyi sanki soru cümlesi değilmiş gibi herhangi bir ekleme yapmadan TİD’e aktarmaktır. Fakat burada yüz ifadesinin önemi ortaya çıkmaktadır. TİD kullanıcısı bu yöntemi kullandığında yüz ifadesi ile soru anlamını vermek durumundadır. Yüz ifadeleri TİD’de oldukça önemlidir ve ilk yöntemin kullanımında da önem taşır. Fakat ikinci yöntem söz konusu olduğunda soru anlamı direkt yüz ifadesinden gelmektedir.



Şekil 2.4 Soru parçacığı [1]

Bir diğer soru cümlesi çeşidi ise soru kelimeleri ile sorulabilen sorulardır. Bu tip soru cümlelerinde soru anlamı cümle içinde yer alan soru kelimesi üzerinden cümleye katılır. Ne, nerede, kaç, ne zaman gibi soru kalıplarının TİD’de sanki farklı bir kelimeymiş gibi başlı başına karşılıkları vardır.

Türkçe’de hal kategorisi kelimelere eklenen -de, -den gibi ekler ile sağlanır. Fakat yapılan araştırmalara göre TİD’de hal belirteçleri bulunmamaktadır [19]. Örneğin “Ben bakkala gidiyorum.” Cümlesinin TİD karşılığı “Ben bakkal gitmek.” Şeklinde olacaktır. Burada dikkat çeken konu, Türkçe’de önemli olabilen tüm son eklerin, TİD’de bir karşılığının olmamasıdır. Dolayısıyla çeviri yaparken kelimeyi eklerinden ayırdıktan sonra, karşılığı olmayan ekleri de göz ardı etmek gerekmektedir.

TİD’de sözcük türetmek iki farklı metotla mümkündür. Bunlardan birincisi, bir nesnenin sahip olduğu algısal imajdan yola çıkarak sözcük türetmektir. Mesela Şekil 2.5’te gösterilen peribacası sözcüğünün türetilmesinde peribacalarının algısal imajından yararlanılmıştır [19].



Şekil 2.5 TİD'de Peribacası [19]

Diğer sözcük türetme yöntemi Türkçe'deki yöntemle daha çok benzer. Bu yöntemde, kaynak sözcük ile türetilen sözcük arasındaki kelime benzerliği olması gerekir. Eğer iki kelimenin de yazılımı arasında benzerlik varsa, TİD karşılığı açısından da bağ kurularak türetme yapılabilir. Mesela "kuvvet" sözcüğü ve "Kuveyt" sözcüğü arasında yazım açısından benzerlik vardır. Bu benzerlikten yararlanılarak sözcükler arasında bağlantı kurulmuş ve "kuvvet" sözcüğünden "Kuveyt" sözcüğü türetilmiştir.

Amerikan İşaret Dili'nde 6000 civarı kelime bulunmasına rağmen, TİD'de 750 civarı kelime bulunduğu bilinmektedir. Dolayısıyla TİD'de, Türkçe'de bulunan her kelimenin bir karşılığı olduğu düşünülmemelidir. Karşılığı olmayan kelimeler parmakla heceleme yapılarak ifade edilebilirler. TİD'de her harfin bir karşılığı bulunmaktadır. Bu harflerden yararlanılarak karşılığı olmayan kelimeler kolayca harf mertebesinde anlatılabilirler.

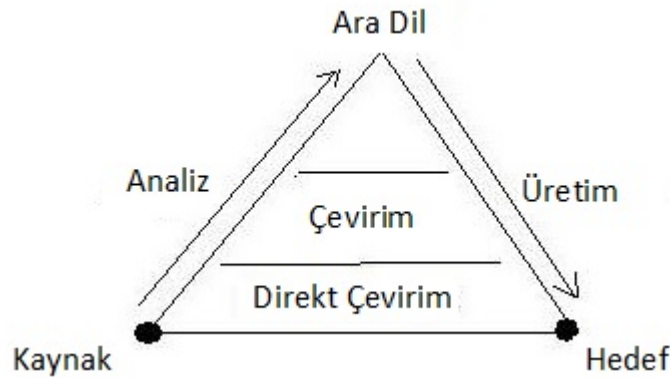
3. BİLGİSAYARLI ÇEVİRME TEKNİKLERİ

Tasarımı yapılan sistem, giriş olarak Türkçe metni alıp, çıkış olarak TİD videoları vermektedir. İki farklı dil arasında çeviri yapabilmek için BÇ yöntemleri kullanılmaktadır. BÇ sistemlerinde, çevirisi yapılmak üzere sistemin girişinde kullanılan dile kaynak dil, sistem çevirimi yaptıktan sonra sistem çıkışında kullanılan dile de hedef dil denilmektedir. Sistem gereksinimlerini karşılayabilmek için, hedef ve kaynak dillerin özelliklerini iyi analiz ederek en uygun algoritmayı kullanmak gerekir. Bu bölümde BÇ ve kullanılan yaygın teknikler üzerine bilgi verilecektir.

3.1. Bilgisayarlı Çevirme

BÇ bir dilden diğer dile çevrim amacıyla kullanılır. BÇ'de önemli olan en iyi çevirimi insan yardımı olmadan yapabilmektir. Temelde BÇ sistemleri, bilgisayar programlarını, sözlükleri ve dilbilimsel kuralları otomatik çevirim yapabilmek için kullanabilirler.

BÇ sistemi ile yapılan çevirimin kalitesi giriş üzerinde ön işleme yapılarak artırılabilir. Ön işleme sırasında giriş metninin ön ekleri veya son eklerinin çıkarılması, metnin daha küçük parçalara ayrılması gibi işlemler uygulanarak çevirim kalitesinin artırılması söz konusu olabilir. Bu işlemler tamamen kullanılacak yönteme bağlı olarak değişebilir.



Şekil 3.1 Bilgisayarlı Çevirme Piramidi

BÇ'de bazı temel yaklaşımlar vardır. Bu yaklaşımlar Şekil 3.1'de yer verilen BÇ Piramidi üzerinde gösterilmiştir. Bu yaklaşımlardan en eskisi direkt çevirme yaklaşımıdır. Giriş metni, yani kaynak, hedefe, yani çıkış metnine birebir çevrilir.

Bu yaklaşımda giriş metnine anlamsal ya da dilbilimsel bir analiz uygulanmaz. Eğer giriş metninin hedef dildeki karşılığı sistemde varsa çeviri yapılır, yoksa çeviri yapılmaz. Bu yaklaşım uygulama açısından kolay olmasına rağmen eski ve yetersiz kalmaktadır.

Ara dil yaklaşımında ise, direkt çevirmenin aksine ara bir dil kullanılır. Bahsi geçen ara dil, kaynak dil ile hedef dil arasında çevirmeyi kolaylaştıran ara bir form olarak düşünülebilir.

Çevirim yaklaşımında ise kaynak dilden hedef dile 3 adımda ulaşılabilir. İlk adımda kaynak dilden, kaynak dil yönelmeli temsiline çevrilir. İkinci adımda kaynak dil yönelmeli temsil, hedef dil yönelmeli temsile çevrilir. Son adımda ise hedef dile çevrim sağlanır. Çevrim sistemleri, giriş metninin analizini, hedef dil ve çıkış metninin analizini ve adımlar arası geçişleri farklı dilbilimsel kısıtlar altında yapar.

Gözleme dayalı yaklaşımda ise, mümkün olduğunca ham veri kullanılmalıdır. Kullanılan ham verilerin hem kaynak hem de hedef dilde karşılıklarının bilinmesi gerekir. Her yaklaşımın kendine özgü avantajları ve dezavantajları vardır. Fakat bu avantaj ve dezavantajlar, söz konusu olan kaynak ve hedef dillerin kendine özgü kısıtları ve hedefleri göz önünde bulundurularak değerlendirilmelidir. Günümüzde en yaygın kullanılan yaklaşım gözleme dayalı yaklaşımdır. Yapılan çalışmada da bu yaklaşımı destekleyen bir yöntem izlenmiştir.

3.2. Bilgisayarlı Çevirme’de Yaygın Kullanılan Yöntemler

Geçtiğimiz yıllarda birçok grup BÇ ve işaret dillerinin otomatik çevrilmesi üzerine farklı metotlar kullanarak çalışmalarda bulunmuştur. Bu metotlar içinde en yaygın kullanılanlar ve bu alanda çalışma yapan otoriteler tarafından benimsenmiş olanlar İstatistiksel Bilgisayarlı Çevirme (İBÇ), Örnek Tabanlı Bilgisayarlı Çevirme (ÖTBÇ) ve Kural Tabanlı Bilgisayarlı Çevirme (KTBÇ)’dir.

3.2.1. İstatistiksel bilgisayarlı çevirme

Geçtiğimiz 10 yılda İBÇ metodunun kullanılabilirliğine dair önemli artış görülmektedir. Bu metotta sisteme tasarımcı tarafından oluşturulmuş kurallar tanımlamak, sözlük

ya da paralel veri kullanmak yerine, istatistiksel karar verme teorisi ve istatistiksel öğrenmeye dayalı yöntemler kullanılır [6].

İBÇ'de kaynak cümlesi ($f_1^j = f_1 \dots f_j$) hedef cümleye çevrilirken ($e_1^l = e_1 \dots e_l$) Bayes Teoremi kullanılır. Eşleştirme yapılırken, formül (3.1)'de gösterildiği gibi, olası tüm hedef cümleler içinden, olasılığı en yüksek olan hedef cümle seçilir.

$$\hat{e}_1^l = \operatorname{argmax}_{e_1^l} \{ \Pr(e_1^l) + \Pr(f_1^j | e_1^l) \} \quad (3.1)$$

İBÇ'de sistemlerin önce eğitilmesi gerekir. Eğitim için hedef dildeki karşılıkları bilinen cümleler paralel veri olarak sisteme aktarılır. Fakat bu yöntem işaret dilinden çevirmede kullanıldığında iki ana problemle karşılaşılabilir. Bunlardan ilki işaret dili için yeterince geniş paralel veri kaynaklarının bulunmayışıdır. İBÇ sisteminde, sistemi önce yeteri kadar geniş paralel veri ile eğitmek gerekir. Sonra da eğitilmiş sistemi yine paralel veri kaynağından yararlanarak test etmek gerekecektir. İşaret dili ve konulan bir dil arasında paralel veri kaynağı bulmak, konuşulan iki dil arasında paralel veri kaynağı bulmaktan çok daha zordur. Kelime ya da kelime öbeği tabanlı bir gösterim sistemi ile konuşulan dil ile işaret dili arasındaki geçiş sağlanabilir. Bu şekilde bir gösterim sisteminin ana sisteme eklenmesiyle, İBÇ tabanlı yapının eğitilmesi ve testlerinin yapılması kolaylaşabilir. Fakat sisteme giriş verilerini önden işleyen farklı bir kısım daha eklemek gerekebilir [6].

İBÇ uygulamaları için sistemin eğitilmesi veya test edilmesi için hazır yazılımlar kullanılabilir. Bu yazılımlardan en çok kullanılanlar GIZA++ ve Moses yazılımlarıdır [20]. Koehn'in çalışmasında da, sistemin eğitilmesi amacıyla GIZA++ yazılımı kullanılmıştır [21]. Othman da çalışmasında İngilizce metni Amerikan İşaret Dili'ne çeviren bir sistem tasarlamış ve yöntem olarak İBÇ kullanmıştır [9]. Bu çalışmada da hazır yazılım olarak GIZA++ ve Moses kullanılmıştır.

3.2.2. Kural tabanlı bilgisayarlı çevirme

Kural Tabanlı Bilgisayarlı Çevirme (KTBÇ) metodunda, İBÇ'nin aksine sistem istatistiksel olarak eğitilmek yerine kurallar ile tanımlanır. Bu kurallar giriş cümlesinin sözdizimsel olarak işlenmesinde, anlamsal olarak yorumlanmasında ve

hedef dil ile içeriksel olarak eşleştirilmesinde önemli rol oynar. Sistemde önceden tanımlanmış dilbilimsel kurallar, hedef ve kaynak diller arası sözlük ve tanımlı kuralları işleyebilecek programlar olması gerekir. Kural tabanlı sistemler, diğer metotların kullanıldığı sistemlere göre daha eskidir.

Kural tabanlı sistemlerde kaynak cümleden hedef cümleye direkt bir çevrim olmaz. Bazı çalışmalarda giriş cümlesinin önce parçalanarak cümle parçacıklarına ayrıldıkları da görülmüştür. Cümle parçalarının öncelikle anlamsal olarak analiz edilmesi gerekir. Bazı sistemlerde detaylı biçimsel analizler de yapılabilir. Bu analizlerden sonra, sisteme önceden tanımlanmış kurallar çerçevesinde kaynak dilden hedef dile çeviri yapılır.

Kural tabanlı sistemler dile bağımlı ya da dilden bağımsız olabilir. Dile bağımlı sistemlerde kaynak ve hedef dillerin bilinmesi önemlidir. Bu durumda kurallar kaynak ve hedef dillere özel olarak üretilir. Kural üretmek ve sisteme eklemek oldukça maliyetli olabilir. Fakat sistem sadece önceden belirtilen diller için tasarlandığından dolayı sistem başarısının daha iyi olması beklenebilir. Dilden bağımsız olarak tasarlanan sistemlerin avantajı kaynak ve hedef dillerin önceden bilinmesine gerek olmayışı ve birden fazla dil üzerinde sistemin çalışabilmesidir. Fakat sistemde olması gereken analiz süreci dile bağımlı sistemlere göre daha detaylı olmalıdır. Ayrıca önceden belirlenmiş kaynak ve hedef diller için eniyileme yapılmamış olması sistem başarısı üzerinde olumsuz rol oynayabilir. Özellikle hedef dil bir işaret dili olduğunda, yazılı bir metinden görsel bir dile eşleşme yapılması gerektiği için, yapılması gereken analiz ve çevrim işlemleri bazı ek konular da göz önünde bulundurularak yapılmalıdır. Bu durumda kuralların hedef ve kaynak dillere bağımlı olarak üretilmesi ve analiz işleminin sadece bu dillerin dilbilimsel özellikleri göz önünde bulundurularak yapılması daha verimli ve daha basit bir sistem tasarlamak adına uygun olabilir.

Örneğin, Türkçe bir cümle Şekil 3.2'deki gibi dilden bağımsız olarak ayrıştırılabilir. Ayrıştırma işlemi tamamlandıktan sonra cümlelerin herhangi bir hedef dile çevrilmesi sisteme önceden tanımlanan kurallar çerçevesinde mümkündür. Şekil 3.2'de "Okul 2 ay sonra bitecek." cümlesi ayrıştırılmış ve ayrıştırılan parçacıklar anlamsal görevlerle eşleştirilmiştir. Sistemde anlamsal görevlere göre kurallar tanımlanabilir. Örneğin hedef dil İngilizce ise "if <future tense> then <add will

before verb>” şeklinde bir kural eklenebilir. Böylece eğer cümlenin zamanı gelecek zaman ise, hedef dilde yüklemden önce “will” kelimesinin eklenmesi kural olarak tanımlanır. Benzer şekilde kurallar tanımlanarak çevirinin yapılması hedeflenebilir.

Okul 2 ay sonra bitecek.



yüklem	bitmek
zaman	gelecek zaman
olumlu/olumsuz	olumlu
kim/ne	okul
ne zaman	2 ay sonra

Şekil 3.2 Örnek bir cümlenin örnek bir KTBC senaryosunda ayrıştırılması

İBÇ ile karşılaştırıldığında, KTBC yönteminde giriş cümlesinin anlamsal ve biçimsel analizi ve sisteme kural eklemek zor bir süreç olarak kalmaktadır. Fakat İBÇ yönteminde de sistemin verimli çalışabilmesi için yeterli miktarda verinin elde edilmesi gerekir.

Bu metot ile yapılan çalışmaların en önemlilerinden bir tanesi Veale'nin (1998) tasarımını yaptığı Zardos isimli sistemdir [4]. Bu sistem İngilizceyi birden fazla işaret diline çevirebilecek şekilde tasarlanmıştır. Sistemde yapay zekâ da kullanılmıştır. Kullanılan işaretlerin gösterimi için bir ara dil kullanılır. Bu arada dilde sadece el hareketleri değil, yüz hareketleri ve diğer hareketlerin gösterimi de yer alır. Sisteme girilen cümle önce parçalara ayrılır ve analiz yapılabilmesi için kısaltılır. Parçalar analiz edilir ve önceden tanımlanan kurallar ve özellikler göz önünde bulundurularak çevrim yapılır. Sistemde çıktı olarak avatar oynatılır.

KTBC yöntemi kullanılarak yapılan önemli çalışmalardan bir tanesi de Marshall ve Sáfár tarafından tasarlanan ViSiCAST (2002) isimli sistemdir [5]. Bu çalışmada İngilizce metinlerin, İngiliz İşaret Dili'ne dönüştürülmesi amaçlanmıştır. Sisteme girilen İngilizce cümleler, sözdizimsel bağımlılığa göre incelenerek ayrıştırılırlar. Ayrıştırılan parçacıklar analiz edilerek HamNoSys adı verilen bir ara dil aracılığıyla

ifade edilir. Bu ara dile çevrim dilbilimsel kurallar çerçevesinde yapılır. Sonrasında da HamNoSys gösterim biçiminden yararlanılarak çıkışta animasyonlar üretilir.

Başka bir çalışmada ise Yunancadan Yunan İşaret Dili'ne çevirme hedeflenmiştir [12]. Yapılan çalışmada sisteme girdi olarak gelen Yunanca metin analiz edilerek ayrıştırılmıştır. Sonra ayrıştırılan parçalar önceden sisteme tanımlanmış kurallar çerçevesinde incelenerek Yunancadan Yunan İşaret Dili'ne eşleştirme işlemi yapılmıştır. Bu eşleme sonunda sistemde çıktı olarak avatar üretildiği görülmektedir.

Bu çalışmalarda dikkat çeken husus, İBÇ yönteminde sistemin eğitilmesi için hazır yazılım gereksinimleri olmasına karşın, KTBC yönteminde de cümle ayrıştırma ve kaynak dilden hedef dile eşleştirme tarafında hazır yazılımlara ihtiyaç duyulmasıdır. Bir diğer konu, kaynak ve hedef diller arasında bir gösterim biçimine veya bir ara dile ihtiyaç duyulmasıdır. Gösterim biçimi kullanılması özellikle animasyon üretimi açısından sisteme otomasyon anlamında çözüm sağlamaktadır. Fakat bu gösterim şekillerinin el hareketleri haricindeki yüz ve benzeri hareketlerde yeterince başarılı olup olmadığı tartışma konusudur. Özellikle TİD gibi yüz işaretlerinin önemli olduğu işaret dillerinde gösterim biçimleri yetersiz olabilmektedir. KTBC yöntemi diğer yöntemlere göre kaynak ve hedef dillerin özelliklerine uygun kurallar çerçevesinde çevrim yaptığı için giriş cümlesinin dilbilimsel olarak ayrıştırılmasını ve incelenmesini gerektirir. Ayrıca kural uygulanması açısından zor bir yöntem olarak görülür. Fakat diğer yöntemlere göre her iki dilin özellikleri göz önünde bulundurularak özelleştirildiği için, diğer yöntemlere göre daha az veri ile daha iyi sonuçlar verebilir.

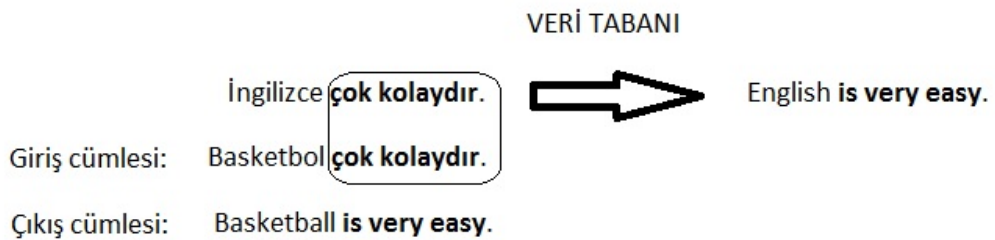
3.2.3. Örnek tabanlı bilgisayarlı çevirme

Örnek Tabanlı Bilgisayarlı Çevirme (ÖTBÇ) yönteminin temelinde, sistemde kayıtlı bir örneğin çeviri amacıyla yeniden kullanılması fikri yer alır. ÖTBÇ tabanlı bir sistemde kaynak dilde birçok cümle ve o cümlelerin hedef dildeki çevirileri yer almalıdır. Bu veri tabanı kullanılarak farklı bir cümlenin hedef dile çevrilmesi sağlanır.

Çevrim sırasında kullanılacak olan kayıtlı cümlelerin, çevrilecek cümle ile yeterli benzerliği sağlaması gerekir. En çok benzerlik gösteren cümle ile giriş cümlesinin eşleşmesi sağlanır. Bu eşleşmenin doğru yapılabilmesi için veri tabanında yer alan kaynak dilde ve hedef dilde cümlelerin birbirleri arasındaki ilişki ve bağlantıların sistemde belirgin olması önemlidir. Sisteme girilen cümle önce dilbilimsel olarak incelenip uygun şekilde parçalara ayrılabilir ve her parça ayrı olarak sistemdeki farklı bir parça ile eşleştirilebilir. Bu parçaların ya da direkt cümlelerin eşleştirilmesinde farklı formüllerin kullanıldığı da görülür. Bu formüllerle cümlelerin ya da cümle parçalarının birbirine yakınlığı ölçülebilir ve eşleşme de elde edilen sonuçlara göre yapılabilir. Bir diğer yöntem kaynak ve hedef cümlelerin karakter bazında eşleştirilmesidir.

Sistemdeki veri tabanını doğru kullanabilmek için, kayıtlı cümlelerin hangi parçalarının çevirilerde tekrar kullanılacağına belirlenmesi önemlidir [22]. Giriş cümlesi veri tabanındaki doğru cümle veya cümle parçaları ile eşleştirildikten sonra bu parçalara karşılık gelen çevirilerin birleştirilmesi de sistemin çıkışında görülecek cümlelerin anlamlı olması açısından önemlidir. Birleştirme yapıldıktan sonra ortaya çıkan hedef cümlelerin bütünlüğü ve anlamsal yeterliliği gözden geçirilmelidir.

Örneğin, Şekil 3.3'te verilen örnekte basit bir giriş cümlesinin veri tabanında bulunan benzer bir cümle ile eşleştirilmesi gösterilmiştir. Giriş cümlesi olan "Basketbol çok kolaydır." ile veri tabanında kayıtlı olan "İngilizce çok kolaydır." cümlelerinin benzerlikleri sistem tarafından algılanabilir. Bu iki cümle eşleştirildikten sonra, veri tabanındaki cümlelerin çevirisi, giriş cümlesinin çevirisi olarak eşleştirilir ve çıkış cümlesi olacak şekilde uyarlanır.



Şekil 3.3 Örnek bir cümle için örnek bir ÖTBÇ senaryosunda eşleştirilmesi

ÖTBÇ yöntemi kullanılarak yapılan çalışmalardan birinde Arapça metinden, Arap İşaret Dili'ne çeviren sistem tasarımı yapılmıştır [11]. Arapça'nın ön ek ve son ek tabanlı bir dil olmasından dolayı kural tabanlı sistemlerin verimli olamayacağı ön görülerek örnek tabanlı bir çalışma yapılmıştır. Sistem giriş cümlesini veri tabanında bulunan parçalarla eşleştirir. Eşleştirme yapılırken giriş cümlesindeki kelimeler teker teker veri tabanında bulunan parçacıklarla kıyaslanır. Benzerlik sağlayan parçacıklar kullanılmak üzere ayrılır. Eşleştirme yapıldıktan sonra çeviride kullanılacak olan parçalar birleştirilir ve ortaya çıkan cümlenin bütünlüğü gözden geçirilir. Birleştirme yapılırken, eşleşen parçacıklarda yer alan kelimelerin sıralamasının düzenlenmesine özen gösterilir.

Başka bir çalışmada ise eşleştirme algoritması olarak dinamik programlama yöntemi kullanıldığı görülür [23]. Diğer yöntemlerde giriş cümlesine en yakın cümlenin tespit edilebilmesi için kelime ya da karakter bazında benzerliğe bakılırken, bu yöntemde anlamsal benzerliğe de bakılmaktadır. Dinamik programlama yönteminde giriş cümlesi ile karşılaştırılan cümle arasında bir uzaklık metriği hesaplanır. Bu uzaklık cümledeki farklı ve aynı kelimelerin sayısının yanı sıra, benzer olan kelimelerin sayısını da hesaba katar. Benzerliklerin dereceleri katsayılar üzerinden değerlendirilir.

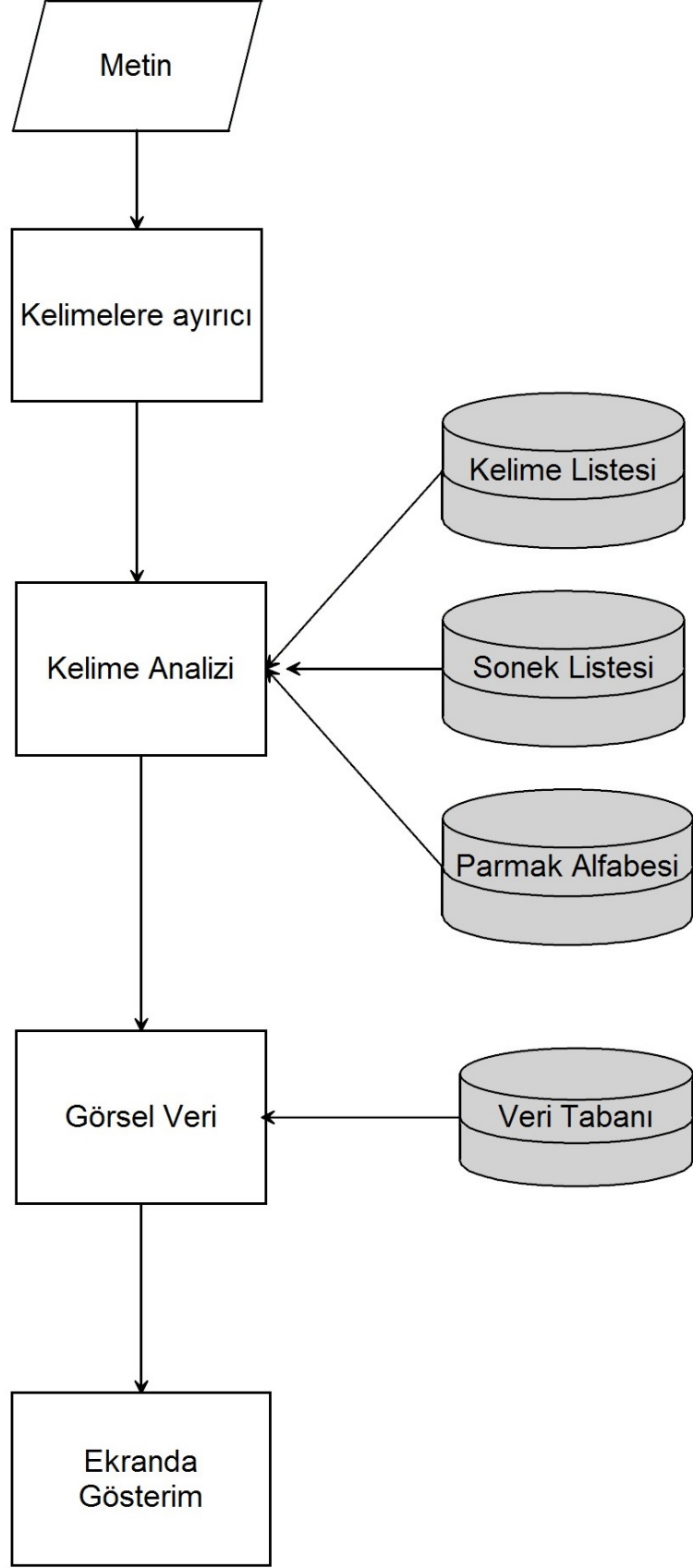
ÖTBÇ yöntemi kural tabanlı yöntemlere göre dilbilimsel inceleme gereksinimini daha az barındırır. Dilbilimsel açıdan analiz yapılması daha zor olan dillerde ÖTBÇ yönteminin kullanılması daha verimli olabilir. Ayrıca İBÇ yöntemine göre sistemin eğitime gereksinimi de bulunmamaktadır. Fakat eşleştirme algoritmasının doğru çalışması sistemin verimliliğini birinci derecede etkiler. Ayrıca cümle tabanlı inceleme yapıldığı için sistemdeki veri tabanının yeterli büyüklükte olması sistem verimliliği açısından önemlidir.

4. YAZILI METİNDEN TİD'e DÖNÜŞTÜREN SİSTEMİN YAPISI

Bilgisayarlı Çevirme yöntemleri Bölüm 3'te açıklandı. Fakat bu çalışmada her ne kadar cümle parçalara ayrılıp kelime üzerine dilbilimsel analiz yapılırsa da, tam olarak Bölüm 3'te değinilen yöntemlerden biri kullanılmamaktadır. Bu yöntemlerin çoğu çalışmada kullanıldığını görülse de, bazı çalışmalarda bu yöntemlerin bir arada kullanıldığı da görülmektedir [10]. Popüler yöntemlerin tamamlayıcı olarak, kaynak ve hedef dillerin niteliklerine uygun bir şekilde bir arada kullanılması, tek bir yöntem ile ilerlemekten daha iyi sonuçlar verebilir [3, 8, 10]. Ayrıca yöntem seçiminde Türkçe ve TİD dilbilimsel özelliklerine uygun olmasına dikkat edilmelidir. KTBC yönteminin bu çalışmada tek başına kullanılması oldukça zor olabilir. Çünkü kural tabanlı yöntemi uygulayabilmek için giriş metninin morfolojik analizinin yapılması gerekir. Ayrıca kaynak ve hedef dildeki kuralların sisteme tanıtılması gerekmektedir. ÖTBC yöntem ise cümle tabanlı çalışmak daha kolay olabilir, fakat genel amaçlı ve esnek olarak tasarlanması amaçlanan bir sistemde sadece cümle tabanlı çalışmak sistemin esnekliği açısından yeteri kadar kullanışlı olmayabilir. İBÇ yönteminde ise sistemin verimli kullanılabilmesi için yeterli büyüklükte iki dilli veri tabanına ihtiyaç vardır. TİD'de ise toplam kelime sayısı 750 civarındadır. Bu tip bir yöntemin bu uygulamada kullanılabilmesi için iyi veri toplamak önemlidir.

Şekil 4.1'de de görüldüğü gibi, yapılan çalışmada kullanılan yöntemde metin kelimelere ayrılarak incelenir. Daha sonra kelimeler analiz edilir ve doğru görsel veri ile bulunur. Son aşamada da görsel veri ekranda gösterilir.

Cümlelerin kelimelere ayrılarak işlenmesinde Türkçe'nin sondan eklemeli bir dil olmasının önemi büyüktür. Kelimelerin analiz edilmesi bakımından kullanılan yöntem kural tabanlı yöntem ile benzerlik göstermektedir. Veri tabanındaki kelimelerle yapılan eşleşmede kullanılan algoritma açısından da kullanılan algoritmanın örnek tabanlı yöntemle ilişkili olduğu görülebilir. Fakat yapılan çalışmada kullanılan BC yöntemi kural tabanlı ve örnek tabanlı yöntemlerin harmanlanmasıyla oluşturulan özgün bir yöntem niteliği taşır. Yapılan çalışmada da özgün bir yöntem kullanılmıştır. Bu çeviride kullanılan yöntemin detaylarına bu bölümde değinilmektedir.



Şekil 4.1 Sistem Blok Diyagramı

4.1. Kelime Ayırıcı

Sisteme girilen metin ilk önce kelime ayırıcı kısmına girerek kelimelere ayrılır. Böylece giriş metninin kelime bazında incelenebilmesine olanak verilir. Kelime ayırıcı metinde yer alan kelimeleri kelimeler arasındaki boşluktan ayırt edebilir. Kelime ayırıcının doğru çalışabilmesi için kullanıcının giriş metnini oluştururken kelimeler arası bir karakter boşluk bırakması önemlidir. Metin kelimelere ayrıldıktan sonra kelimeler analiz edilmek üzere bir sonraki kısma aktarılır.

Örneğin “ “eren” 3 gün sonra taşınıyor” cümlesi sisteme girildiğinde, kelime ayırıcının çıkışında bu cümleyi oluşturan kelimeler ayrıştırılmış ve kelime analizine hazır olacaktır. Bu cümlenin kelimelere ayrıştırılması Çizelge 4.1’de gösterilmiştir.

Çizelge 4.1 Kelime Ayırıcı Örneği

GİRİŞ	ÇIKIŞ
“eren” 3 gün sonra taşınıyor	“eren”
	3
	gün
	sonra
	taşınıyor

Her ne kadar örnek tabanlı ya da kural tabanlı çalışmalarda giriş cümlesinin parçalara ayrıldıktan sonra ayırık olarak incelendiği görülse de, metni kelimelere ayırıp incelemek cümlenin bütünlüğünün korunması açısından yeterli değilmiş gibi görünebilir. Fakat her ne kadar Türkçe sadece kelime ve anlam bütünlüğü bakımından çok zengin bir dil de olsa, TİD anlam bütünlüğü açısından çok yeterli değildir. TİD’de cümlelerin anlamı genelde kelimelerdeki anlamların bir araya eklenmesi ile ortaya çıkar. Türkçe kelimelerin sonundaki eklerin çoğu TİD’de karşılık bulmamaktadır. Bunlar göz önünde bulundurulduğunda cümleyi kelimeler bazında ele almak, kelimeleri iyi analiz etmek mantıklı olabilir.

4.2. Kelime Analizi

Giriş metni kelimelere ayrıldıktan sonra bu kelimelerin analiz edilmesi gerekir. Bu kısımda kelimenin birkaç farklı sınıftan hangisine ait olduğunun tespit edilmesi ve çevrimin buna göre yapılması hedeflenmiştir.

Kelime analizi yapılırken kelimenin öncelikle direkt olarak veri tabanındaki kelimelerden bir tanesi ile eşleşip eşleşmediği kontrol edilir. Direkt eşleşmesi beklenen kelimeler herhangi bir ek almadan saklandıkları için veri tabanında “kök” klasörü altında bulunurlar. Direkt eşleşen bu kelimeler sistemde “kök” olarak etiketlenirler.

Eğer kelime kök değilse, sistem bir sonraki adımda kelimenin özel isim olup olmadığını kontrol eder. Giriş metninde yer alan özel isimlerin başında ve sonunda tırnak işareti (“) yer almalıdır. Eğer kelimenin özel isim olduğuna karar verilirse, bu kelime harflere ayrılarak parmakla heceleme yöntemi ile çevrilecektir. Aynı yöntem daha sonra veri tabanında olmayan ya da eşleştirelemeyen kelimeler için de uygulanacaktır. TİD’de kelime sayısı oldukça düşüktür ve Türkçe’deki her kelimenin bir TİD karşılığı bulunmamaktadır. Bu yüzden eşleştirme yapılamayan kelimelerin de özel isimler gibi parmakla hecelenerek TİD’e çevrilmesi hedeflenmiştir. TİD kullanıcıları da günlük hayatta karşılığı olmayan ya da bilmedikleri kelimeleri bu yöntemle çevirdikleri için sistem bu açıdan tüm kelimeleri çevirebilecek özellikte sayılabilir.

Eğer kelime kök ya da özel isim değilse, kelimenin işlenmesi faydalı olacaktır. İşleme sırasında kelimenin sonek alıp almadığına, sonek varsa bu sonekin TİD için anlamlı olup olmadığına bakılması gerekir. Kelimenin soneklerden ayrılıp kökünün tespit edilmesi, veri tabanındaki kelimelerle karşılaştırıp eşleştirme yapabilmek açısından önemlidir.

Kelime ayrıştırıcı örneğinde cümle kelimelere ayrıldıktan sonra kelimelerin etiketlenmesi gerekir. İlk kelime olan “eren” tırnak içinde yazıldığı için özel isim olarak etiketlenir. Sonraki kelimeler “3”, “gün” ve “sonra” kelimeleri veri tabanında kök dosyası altında yer alan kelimelerden biri ile direkt eşleşme sağladıkları için kök olarak etiketlenirler. Fakat son kelime olan “taşınıyor” özel isim değildir. Ayrıca

veri tabanındaki kelimeler ile de direkt eşleşme sağlamadığı için kök olarak nitelendirilemez. Bu durumda sistem bu kelimeyi işlenecek olarak nitelendirir ve bu şekilde etiketler. Çizelge 4.2’de örnek cümledeki kelimelerin nasıl etiketlendiği gösterilmiştir.

Çizelge 4.2 Kelime etiketleme örneği

Kelimeler	Kök	Özel İsim	İşlenecek
eren		eren	
3	3		
gün	gün		
sonra	sonra		
taşınıyor			taşınıyor

4.2.1. Sonek bulma

Türkçe sondan eklemeli bir dil olduğu için, kelimelerin cümle içinde kazandığı anlamın tamamlanabilmesi için kelimenin soneklerinin tespit edilmesi önemlidir. Sistem işlenmesi gereken bir kelimenin öncelikle son ekini kontrol eder ve kelimedenden ayırır. Kelimenin ek ayrıldıktan sonraki hali gövde olarak adlandırılır.

Soneki tespit etmek için sistem kelimeyi sondan incelemeye başlar. Tespit edilen sonekin anlamlı olup olmadığını anlamak önemlidir. Veri tabanında ayrı bir klasör altında anlamlı olabilecek sonekler yer alır. Anlamlı olmasından kasıt, bahsi geçen sonekin TİD tarafında bir anlamsal karşılığının olmasıdır. Eğer tespit edilen sonek anlamlıysa, kelimenin kökü tespit edildikten sonra sonek kökün sonuna sanki farklı bir kelimeymiş gibi eklenir.

Örneğin, “çalışıyorum” sözcüğünde “-yorum” ekinin TİD tarafındaki anlamsal karşılığı birinci tekil şahıs anlamı katıyor olmasıdır. Bu durumda bu kelimenin TİD karşılığı aslında “çalışmak + ben” ya da “ben + çalışmak” şeklinde olacaktır. Bu kelime soneki bulunmak üzere incelenmeye başlandığında son harfinden başlanarak mümkün olan bütün olasılıklar ek olabilecek şekilde değerlendirilir. Tüm olasılıklar ayrı bir dosya altında kayıtlı bulunan, anlamlı sonekler ile karşılaştırılır. Bu soneklerden bir veya daha fazlası ile eşleşme yapılabilir. Fakat

bu eşleşmelerden en çok harf ile olanı dikkate alınacaktır. Eğer “çalışıyorum” kelimesinin sonek analizinde “-yorum” ekinin haricinde “-ıyorum” ya da daha fazla harfli bir ek de eşleşme sağlasaydı “-yorum” yerine dikkate alınacaktı. Fakat bu kelimedede en yüksek harfli eşleşme, Çizelge 4.3’te gösterildiği gibi “-yorum” ekinde sağlandığı için “-yorum” sonek olarak alınmıştır.

Çizelge 4.3 Sonek bulma örneği

İŞLENECEK KELİME	SONEK ANALİZİ
çalışıyorum	m
	um
	rum
	orum
	yorum
	ıyorum
	şıyorum
	ışıyorum
	lışıyorum
	alışıyorum
	çalışıyorum

Analizde karşılaştırılan diğer olasılıklardan da sonek olanlar çıkabilir. Fakat önemli olan ilgili sonekin TİD tarafında anlamlı olarak nitelendirilmiş olmasıdır. Anlamlı olabilecek sonekler ayrı bir dosya altında tutulmaktadır. Böylece anlamsız olan ekler eşleşme sağlamayarak elenecektir.

4.2.2. Eşleştirme algoritması

Kelimenin soneki olup olmadığı kontrol edildikten sonra kelime sonekenden ayrılmış olarak bir sonraki aşamaya geçer. Bu aşamada kelime kök halinde bulunabilir. Eğer ekten ayrıldıktan sonra kelime kök haline gelirse bu bir sonraki aşama için en iyi durum olarak nitelendirilebilir. Çünkü kök halindeki bir kelimenin veri tabanında olup olmadığını tespit etmek daha kolaydır. Fakat her zaman sonek ayrıldıktan sonra kelime kök halinde olmayabilir. Veya kelimenin soneki anlamlı değilse sonek tespit edilememiş ve kelime cümlede bulunduğu ekli hali ile

bulunuyor olabilir. Bu durumda kelimenin veri tabanında olup olmadığını tespit etmek için kelimenin kökünün tespit edilmesi gerekir.

Tasarlanan eşleştirme algoritması ilgili kelimeyi veri tabanında kök olarak kayıtlı olan tüm kelimelerle karşılaştırır. Her karşılaştırma sonunda bir uzaklık değeri üretilir. Bu uzaklık, ilgili kelimenin veri tabanındaki kelime ile olan uzaklığını göstermektedir. Bu uzaklık değeri gövde halindeki kelimedenden kök halinin çekilebilmesi için önemlidir.

Yapılan çalışmada bu uzaklık değerinin hesaplanması için kullanılan uzaklık formülü dinamik programlama (DP) algoritmasında kullanılan uzaklık formülünden esinlenilerek türetilmiştir. Bu formül (4.1)'deki gibi tanımlanır. Çizelge 4.4'te bu formülde bulunan simgeler ve açıklamaları yer almaktadır. DP, ÖTBÇ uygulamalarında 1990'ların başından bu yana kullanılan bir algoritmadır. Sumita ÖTBÇ uygulamasında DP algoritması kullanmış ve oldukça verimli olduğunu tespit etmiştir [23]. Bu uygulamada giriş cümlesi ve veri tabanındaki cümleler arasındaki uzaklık iki cümlede yer alan kelimelerin karşılaştırılması ile belirlenir.

$$uzaklık = \frac{E + \text{Ç} + 2 \sum AU}{B_{giriş} + B_{örnek}} \quad (4.1)$$

Çizelge 4.4 DP Uzaklık Denklemi Parametreleri

DP Uzaklık Denklemi Parametreleri	Açıklama
uzaklık	Giriş cümlesi ve örnek cümle arasındaki uzaklık
E	Eklenmiş kelime sayısı
Ç	Çıkarılmış kelime sayısı
AU	Kelimeler arası anlamsal uzaklık
$B_{giriş}$	Giriş cümlesinin uzunluğu
$B_{örnek}$	Örnek cümlelerin uzunluğu

Uzaklık hesaplanırken iki cümle arasında fazladan bulunan kelimelerin sayısı (E ve Ç) ve aynı sırada ama farklı olan kelimelerin birbirleri ile olan anlamsal uzaklıkları (AU) hesaplanır ve sonrasında giriş cümlesi ile karşılaştırılan cümlelerin

uzunlukları ($B_{giriş}$ ve $B_{örnek}$) kullanılarak normalleştirilir. Bu formül iki cümle arasındaki uzaklığı hesapladığı için kelimeler arasındaki anlamsal benzerlik de hesaplamada göz ardı edilmemiştir. Anlamsal uzaklık hesaplanırken eş anlamlılık seviyesi kontrol edilir. Bu işlem için de eş anlamlılar sözlüğünden yararlanılır. Anlamsal uzaklık 0 ile 1 arasında değer alır.

DP algoritmanda hesaplanan uzaklık iki cümle arasında yapılmaktadır. Fakat benzer bir uzaklık hesabının kelimeler arasında yapılması gerekirse, benzer mantıkla harf bazında bir karşılaştırma yapılabilir ve anlamsal uzaklığın hesaplanmasına bu şekildeki bir hesaplamada gerek duyulmaz. Özetle, bu hesaplama kelimeler arasındaki uzaklığın hesaplanmasında kullanılacak olursa bu çalışmada kullanılan (4.2)'deki formül elde edilebilir. Çizelge 4.5'te denklemden kullanılan simgeler ve açıklamaları yer almaktadır. Formüldeki katsayılar deneysel olarak seçilmiş ve yapılan testlerde uygun oldukları görülmüştür.

Yine “çalışıyorum” kelimesini örnek gösterilecek olursa, bu kelimedeki sonek “-yorum” olarak tespit edilir ve kelimedenden ayrılır. Bu durumda kelime “çalışı” haline gelir. Kelime bu haliyle veri tabanında kayıtlı olan “çalış” kelimesi ile eşleşme sağlayamaz. Bu durumda kelimenin veri tabanında kayıtlı olan kök halini bulabilecek esnek bir eşleşme algoritmasına gerek duyulur.

$$u = \frac{2f+e}{2a+2f+e} \quad (4.2)$$

Çizelge 4.5 Uzaklık Denklemi Parametreleri

Uzaklık Denklemi Parametreleri	Açıklama
u	Uzaklık
f	İki kelime arasındaki farklı harf sayısı
e	İşlenen kelimedeki karşılaştırılan kelimeye göre fazladan kaç harf bulunduğunu gösterir
a	İki kelime arasındaki sıralı olarak eşleşen harf sayısını gösterir

Denklemin 4.2'de gösterildiği gibi, iki kelime arasındaki uzaklık (u) kelimelerin harf bazında karşılaştırılmasıyla elde edilir. Bu karşılaştırmada önce karşılaştırılacak

kelimenin veri tabanındaki kelimedenden boyut olarak büyük olup olmadığı kontrol edilir. Eğer karşılaştırma yapılacak kelimenin boyutu büyük ya da eşit ise uzaklık formülü uygulanabilir. Eğer kelime veri tabanındaki kelimedenden büyük ise, iki kelime arasındaki harf sayısı farklılığı formülde “e” ile gösterilir. İki kelime arasında eşleşen harf sayısı (a) ve farklı olan harf sayısının (f) tespit edilmesi önemlidir. Farklı olan harfler ve fazladan bulunan harfler (e) kelimenin farklılığını belirleyecektir. Fakat kelimeler arasındaki farklı harflerin bu uzaklıktaki etkisi, fazladan bulunan harflere göre daha fazladır. Bu da formülde katsayı kullanılarak belirtilmiştir. Bu uzaklık değeri kelimedeki harf sayısına göre normalleştirilir. Sonuçta ortaya çıkan uzaklık değeri 0 ile 1 arasında değişecektir. Tamamen eşleşen bir kelimenin uzaklığı 0 çıkarken, tamamen farklı olan bir kelimenin uzaklığının 1 olması beklenmektedir. Sistem veri tabanındaki kelimeler ile ilgili kelimenin uzaklıklarını karşılaştırır ve en düşük uzaklık değerine sahip olan veri tabanındaki kelime ile eşleştirme yapar.

Yapılan çalışmada ilgili kelime, veri tabanındaki kelimelerde karşılaştırılır ve elde edilen uzaklık değerleri göz önüne alınır. En küçük uzaklık değeri bu algoritmada belirleyicidir ve sistem tarafından en küçük uzaklığa sahip kelime ile eşleştirme yapılır. Fakat eşleşmenin doğru yapılabilmesi için dikkat edilmesi gereken bir diğer konu da bu uzaklık değerinin diğer uzaklık değerlerine göre küçük de olsa yine de anlamlı bir değer olmasıdır. Bu yüzden sistemde belirlenen bir eşik değeri ile tespit edilen en küçük uzaklık değerinin karşılaştırılması gerekir. Eğer en küçük uzaklık değeri eşik değerinden büyük ise veri tabanında uygun eşleşme yapılamayacağı algılanır. Bu eşik değeri de 0 ile 1 arasında herhangi bir değer olabilir. Fakat bu değer 0'a yakın seçilmesi daha fazla kelimenin parmakla hecelenmesine, 1'e yakın seçilmesi ise yapılacak eşleşmelerin anlamsız olabilmesine yol açar. Bu durumda eniyileme yapmak gerekebilir. Yapılan çalışmada deneysel olarak 0.45 değerinin uygun olduğu görülmüştür. Fakat veri tabanı değıştikçe eşik değerinin de değışmesinin faydalı olacağı söylenebilir.

Eğer eşleşme olmayacaksa sistem ilgili kelimenin özel isimlerde olduğu gibi harf bazında parmakla heceler. Bu durum söz konusu ise, öncesinde kullanıcıya kelimenin veri tabanında bulunmadığı ve parmakla heceleme yapılacağına dair

bilgilendirme yapılır. Eğer eşleşme yapılacaksa, bulunan kök ve eğer varsa son ek birleştirilir.

Örneğin “istiyorum” kelimesi işlenecek olursa, kelimedeki ekin “-yorum” olduğu belirlenecek ve kelimedен ayrılarak geriye gövde halinde “isti” kalır. Geriye kalan bu gövde, veri tabanında kayıtlı olan kök halindeki kelimelerle karşılaştırılır. Veri tabanında kayıtlı “iste” kelimesi ile “isti” arasındaki uzaklık diğer kelimelere göre daha az çıkar. Bu iki kelime arasında ilk 3 harf sıralı olarak aynıdır ve uzaklık denkleminde $a=3$ 'tür. Gövde ile karşılaştırılan kelimeler aynı sayıda harften oluştuğu için $e=0$ 'dır. Kelimelerdeki son harf farklı olduğu için $f=1$ 'dir. Dolayısıyla bu iki kelime arasındaki uzaklık $[(2*1)+0]/[(2*3)+(2*1)+0] = 0.25$ 'tir. Diğer kayıtlı kelimelerle “isti” arasındaki uzaklık daha büyüktür. 0.25 değeri, eşik değeri olan 0.45'ten küçük olduğu için “iste” kök olarak alınır. Böylece kelime işlendikten sonra çıkışta “istiyorum” = “iste” + “yorum” şeklinde gösterilir.

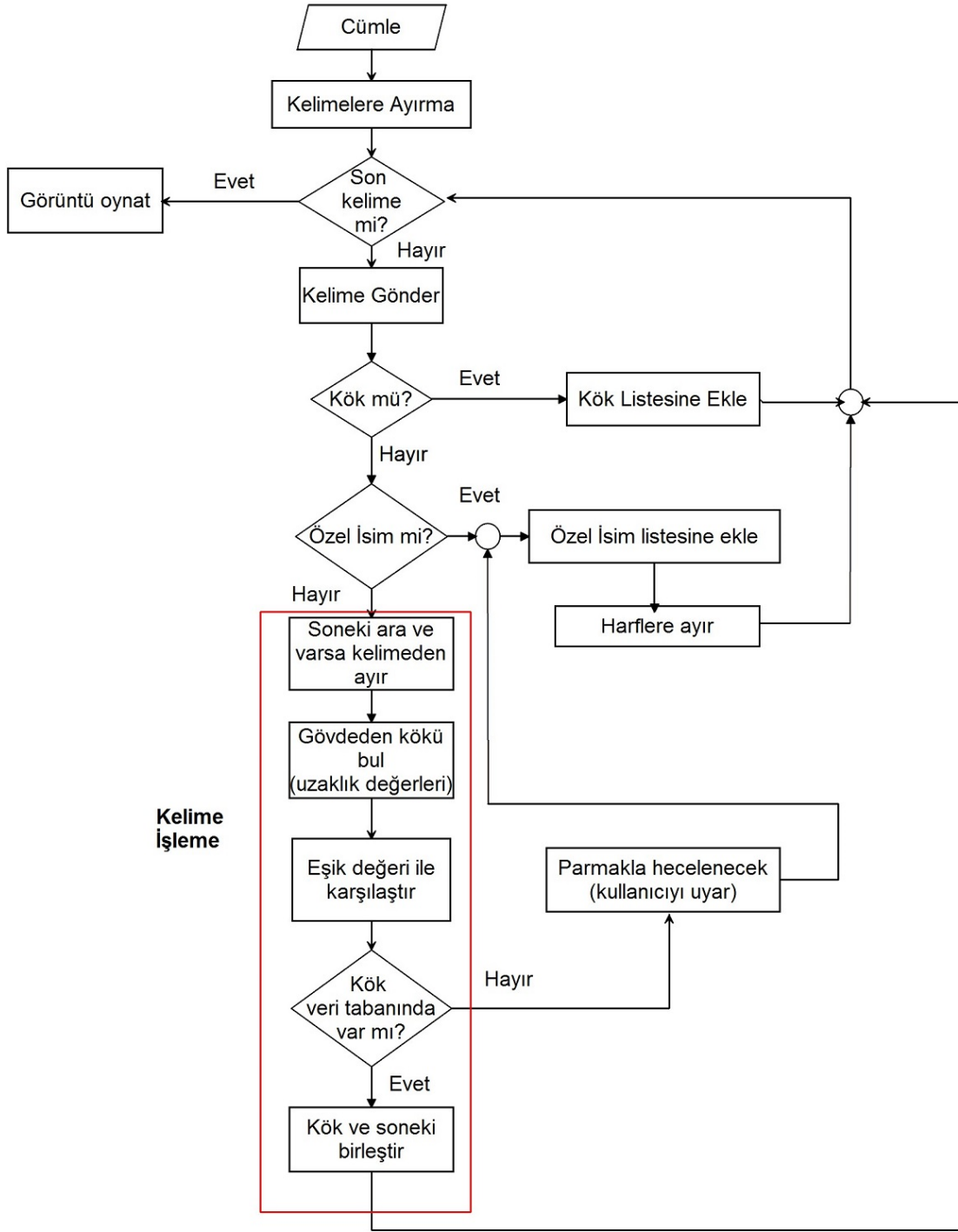
Bir diğer örnekte “indirim” kelimesinin analizi incelenebilir. Bu kelime kök halindedir ve tersten incelenip ek var mı diye kontrol edilse de anlamlı ek bulunamaz ve mevcut haliyle veri tabanındaki kelimelerle uzaklık karşılaştırmasına girer. Veri tabanında bulunan çoğu kelime ile olan uzaklık 1 çıkacaktır. Fakat bazı kelimelerle benzerlikler olacağı için 1'in altında sonuçlar çıkabilir. Örneğin “iyi “ kelimesi ile karşılaştırma yapıldığında sonuç $[(2*2)+4]/[(2*1)+(2*2)+4]=0.8$ 'dir. Bu değer mevcut veri tabanı için veri tabanındaki kelimelerle yapılan karşılaştırmalarda elde edilen en düşük değerdir. Fakat yine de eşik değeri olan 0.45'in altında kaldığından dolayı eşleşme sonucu olarak en düşük uzaklık değerini sağlayan kelimeyi almak yerine, eşleşme yapılmayacak ve kullanıcıya kelimenin veri tabanındaki kelimelerle eşleştirilemediği ve parmakla heceleme yapılacağı yönünde bilgi verilecektir.

Çizelge 4.6 Örnek cümle analizi

KELİME	BULUNAN KÖK	EN KÜÇÜK UZAKLIK	EŞLEŞME DURUMU
yarın	yarın	0	Veri tabanı
ofise	o	0.6	Parmakla heceleme
gelecek	gel	0.4	Veri tabanı
misin	misin	0	Veri tabanı
hastaneye	hafta	0.33	Veri tabanı
gidiyor	git	0.43	Veri tabanı
musun	musun	0	Veri tabanı

Çizelge 4.6’da verilen örnekte “yarın”, “misin” ve “musun” kelimeleri veri tabanında kayıtlı kelimelerle direkt eşleşme sağlamıştır. “ofise” kelimesi veri tabanında kayıtlı kelimelerle karşılaştırıldığında, en yakın kelime “o” olarak belirlenmiştir. Fakat bu iki kelime arasındaki uzaklık eşik değerinden büyük olduğu için sistem bu iki kelimeyi eşleştirmek yerine, “ofise” kelimesinin parmakla hecelenmesine karar vermiştir. “hastaneye” kelimesinde ise bu kelimeye en yakın eşleşme “hafta” kelimesi ile olmuştur. Bu iki kelime arasındaki uzaklık 0.33’tür. Bu uzaklık değeri eşik değerinden küçük olduğu için sistem “hastaneye” kelimesinin kökü olarak “hafta” kelimesini seçer ve yanlış bir eşleştirme yapar. Eğer veri tabanında kayıtlı “hastane” kelimesi olsaydı, sistem “hastaneye” kelimesinin kökü olarak bu kelime ile eşleşme sağlayacaktı. Veya sistemdeki eşik değeri 0.33’ten küçük olsaydı, bu durumda eşleşme geçersiz sayılacak ve “hastaneye” kelimesi parmakla hecelenecekti.

Sistem cümleyi kelimelere ayırarak ve kelimelerin sınıflandırılarak gerekirse eklerine göre analiz edilmesi açısından kural tabanlı sistemlerle benzerlik göstermektedir. Fakat kelimelerin veri tabanındaki kelimeler ile karşılaştırılması örnek tabanlı uygulamalar ile yöntemsel olarak benzerlik göstermektedir. Dolayısıyla bu sistem yöntemsel olarak karma bir sistem olarak gösterilebilir. Karma yöntemler kullanılarak kaynak ve hedef dillerin özellikleri ile en iyi şekilde eşleşen sistem tasarımı yapılabilir. Şekil 4.2’de tasarlanan sistemin akış diyagramı verilmiştir. Şekil 4.3’te de kelime işleme algoritması sözde kod olarak gösterilmiştir.



Şekil 4.2 Akış diyagramı

Algoritma: Kelime İşleme

```
For (kelime(son(x) harf), x++)
    if (son(x) harf = sonek dosyasında bir dosya ismi)
        Then (sonek = son(x) harf)
    End
End
For (i=0, i=length(kelime) - lenght(sonek), i++)
    Gövde[i] = kelime [i];
End

For (i=0, i=length(kök klasörü), i++)
    Dist[i] = uzaklık(Gövde,kök klasörü[i])
    if(Dist[i] <=others)
        Then(
            min Dist = Dist[i];
            temp kök = kök klasörü[i];
        )
    End

if(min Dist > eşik değeri)
    Then(parmakla hecele)
Else
    Kök = temp kök;
    Yeni kelime = kök + sonek
```

Şekil 4.3 Kelime işleme algoritmasının sözde kodu

4.3. Görsel Veri

Sistem giriş metninde yer alan kelimeleri sınıflandırdıktan ve işledikten sonra TİD karşılıklarını üretmek durumundadır. Çıktı olarak veri tabanında bulunan görsel veriler kullanılır ve ekranda gösterilir.

Kullanılan görsel veriler veri tabanında kayıtlı resimlerin sıralı olarak oynatılması ile elde edilir. Kayıtlı resimlerin hepsi aynı boyuttadır ve hangi resmin hangi kelimeye ait olduğu veri tabanında dosyalama yöntemi ile belirginleştirilmiştir. Aynı kelimenin altında olan resimlerin sistemde kayıtlı isimleri çağrılacakları numaralar şeklinde verilmiştir. Böylece resimleri sıralı olarak çağırarak mümkündür.

Alternatif olarak görsel veri olarak resim ya da video yerine animasyon üretmek de mümkündür. Animasyon üretmek için genel geçer gösterimler ara dil olarak kullanılabilir. Bu gösterimlerden yola çıkarak hazır yazılımlardan faydalanarak animasyon üretilebilir. Animasyon üretmek veri tabanında hazır görsel veri saklama ihtiyacını ortadan kaldırır. Bu da özellikle gömülü uygulamalarda sistem belleği ihtiyacını en aza indirmektedir. Fakat animasyonlar resim ve videolara göre özellikle yüz hareketlerini ifade etme açısından çok yeterli kalmayabilir. Özellikle yüz hareketlerinin baskın olduğu işaret dillerinde çıkış olarak animasyon kullanılması sistemin anlaşılma yüzdesini olumsuz etkileyebilir. TİD'de de yüz hareketleri oldukça önemlidir. Bu durum göz önünde bulundurularak bu çalışmada görsel veri olarak resimlerden elde edilen videolar kullanılmaktadır. Şekil 4.4'te kullanılan resimlerden bir tanesi gösterilmektedir.



Şekil 4.4 Örnek resim

Sistemde kullanılan görsel veri, resimlerin çerçeve olarak kullanılarak video şeklinde gösterilmesi ile elde edilmektedir. Bu yöntem yerine direkt video da kullanılabilir. Eğer sistem sadece yazılım tabanlı olacaksa, video formatının çözülmesi sorun yaratmayabilir. Fakat gömülü sistemlerde direkt video kullanmak

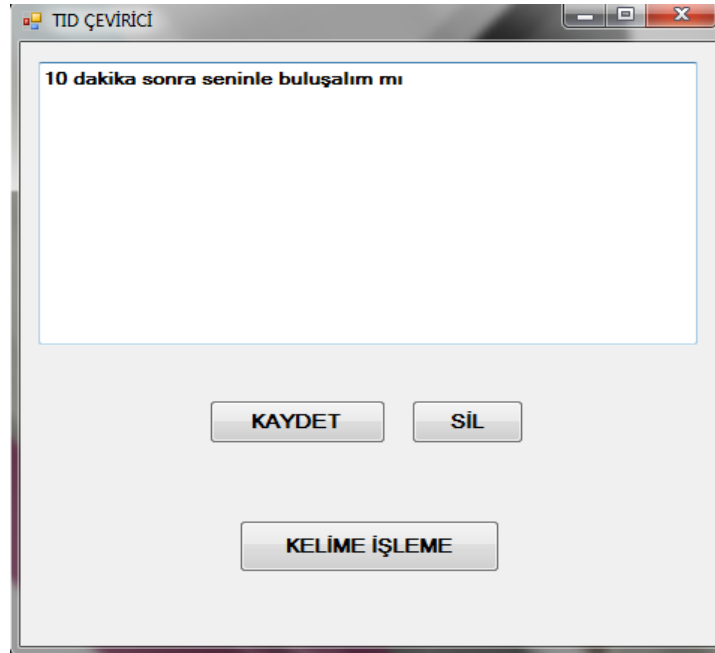
bu açıdan sorun yaratabilir. Bu açıdan resimlerin sıralı gösterimi ile elde edilen görsel veri kullanımı sistem açısından daha uygun görünmektedir.

Sistemde kullanılan görsel veri oluşturulurken, Dikyuva ve Zeshan tarafından yazılan kitapla beraber TİD eğitimi için kullanılan eğitim videolarından yararlanılmıştır [17]. Videolardan resimler yakalanarak uygun formatta düzenlenmiş ve sistemde görsel veri olarak kullanılmıştır. Fakat kullanıcılar sadece bu videolarla sınırlı kalmak durumunda değildir. Sistem görsel veri eklenmesi açısından oldukça esnek tasarlanmıştır. Kullanıcıların sisteme veri eklemeleri için koda müdahale etmeleri gerekmez. Sadece resimlerin uygun büyüklük ve formatta olması ve doğru dizin altına eklenmesi yeterlidir. Sistem ilgili kelime ile eşleşme yapılması durumunda yeni eklenmiş bile olsa ilgili görsel veriyi çıkışta kullanacaktır.

5. UYGULAMALAR

Bölüm 4'te detaylıca anlatılan yapı, Visual Studio programı kullanılarak gerçekleştirilmiştir. Kelime ayrıştırma, kelime analizi ve görsel veri çağrılması gibi bölümler C# tabanlı bir program üzerinden gerçekleştirilmiştir. Yapılan çalışmada önce bilgisayar ortamında yazılım geliştirilmiş ve bilgisayar ortamında çalıştırılmıştır. Tasarım ve yazılım geliştirme tamamlandıktan sonra yazılım WinCE işletim sistemine uyumlu hale getirilerek ARM tabanlı işlemciye gömülmüştür.

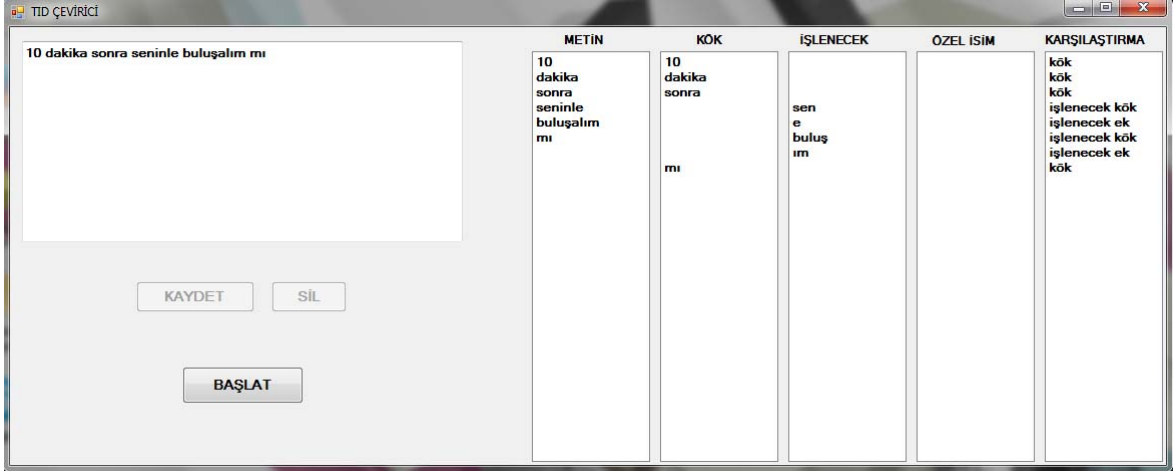
Program giriş metnini program arayüzü üzerinden alır. Kullanıcı arayüzüne Şekil 5.1'de de gösterildiği gibi çevrilmesi istenen metin yazılmalıdır. Metin yazımı tamamlandıktan sonra "KAYDET" butonuna basılarak yazılan metnin sisteme kaydedilmesi sağlanır. Eğer yazılan metin silinmek isteniyorsa "SİL" butonu kullanılabilir ve yazım kutucuğu yeni metin yazımı için hazır hale getirilebilir. Metin yazımı tamamlandıktan sonra kaydedildiğinde kelimeleri ayırmak için kullanıcıdan "KELİME İŞLEME" butonuna basmasını bekler.



Şekil 5.1 Program arayüzü

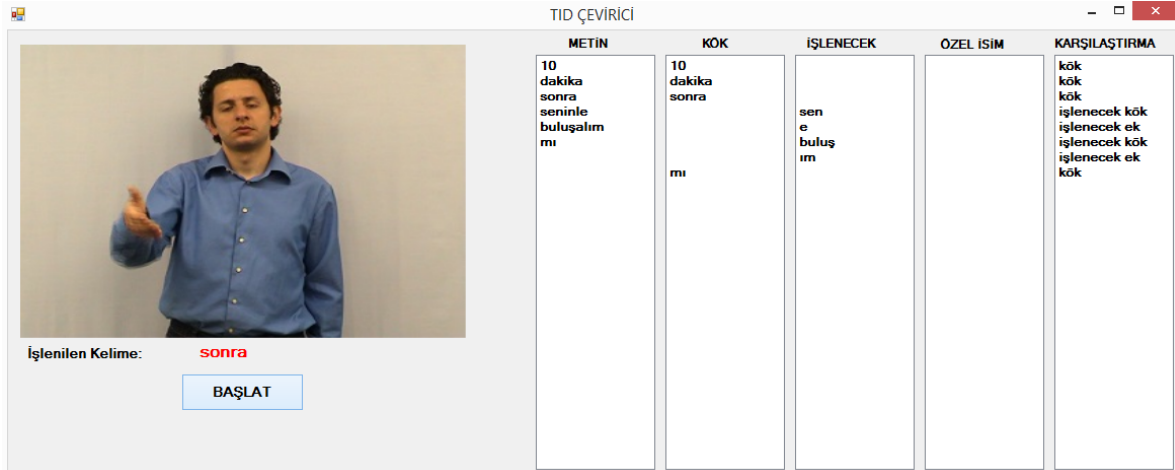
Kullanıcı ilgili butona bastıktan sonra giriş metni Şekil 5.2'de gösterildiği gibi kelimelerine ayrılmış ve işlenmiş olarak program arayüzü üzerinde kullanıcıya sunulur. Kelimeler işlendiğinde Bölüm 4'te anlatıldığı gibi kök, özel isim ya da

işlenecek olarak sınıflara ayrılır ve metin içinde yer alan kelimelerin hangi sınıflarda inceleneceği kullanıcıya gösterilir.



Şekil 5.2 Programda kelime işlenmesi

Kelimeler işlendikten sonra, TİD karşılıklarının gösterilmeye başlanması için kullanıcının “BAŞLAT” butonuna basması beklenir. Sonrasında işlenen kelimelerin TİD karşılıkları Şekil 5.3’te görüldüğü gibi gösterilmeye başlanır. O an hangi kelimenin TİD karşılığı gösteriliyorsa o kelime program arayüzünde yazılı olarak da görünür.



Şekil 5.3 Programda kelimelerin TİD karşılıklarının gösterilmesi

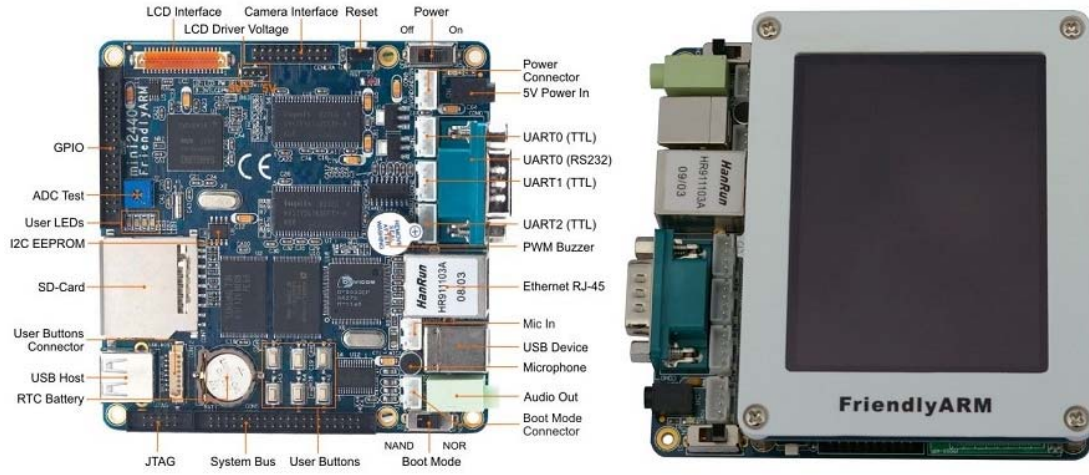
Veri tabanında kayıtlı olan görsel veri ayrı dosyalar altında tutulur. Her kelimeye karşılık gelen görsel veri kök dosyası altında kendi adında ayrı bir dosya altında tutulur. Görsel veri olarak jpeg formatında kaydedilmiş resimler kullanılır. Bu

resimler sıralı çerçeve olarak oynatılır ve video olarak gösterilir. Aynı kelimeyi gösteren resimlerin isimleri hangi sırada gösterileceklerini belirten numaralar olarak verilir. Şekil 5.4'te kelimenin hangi dizin altında nasıl kaydedilmesi gerektiğini gösteren bir örnek verilmiştir. Bu yapıda veri tabanına kelime eklemek için yazılımda herhangi bir değişiklik yapmak gerekmez. Resimleri uygun formatta uygun dizin altına sıralı olarak eklemek yeterlidir. Yazılım yeni eklenecek kelimeleri de tanıyacak esneklikte tasarlanmıştır.



Şekil 5.4 Kelime dizin örneği

Sistemde donanım olarak FriendlyARM Mini2440 geliştirme kiti kullanılmıştır. Şekil 5.5'te bu geliştirme kitinin devre yapısı verilmiştir. Bu donanımda ARM9 tabanlı 400MHz'lik güçlü bir işlemci bulunmaktadır. Kit üzerinde SD Kart soketi ve USB girişi bulunmaktadır. Veri tabanını tutmak için SD Kart veya USB arayüzlü çakarbellek kullanılabilir. Yapılan testlerde USB arayüzlü çakarbellek kullanılmıştır. Ayrıca donanım üzerinde dokunmatik LCD ekran da bulunmaktadır. Bu LCD üzerinde giriş metninin TİD karşılığı olan görsel veri gösterilir. Kullanılan geliştirme kartının donanımsal özellikleri Çizelge 5.1'de verilmiştir.



Şekil 5.5 Kullanılan donanımın devre yapısı [16]

Geliştirme kiti üzerinde işletim sistemi olarak WinCE koşturmaktadır. Sistemde WinCE koşması hem yazılım açısından kolaylık sağlamaktadır, hem de hazırlanan yazılım üst seviye bir uygulama niteliğinde olduğundan WinCE koşan herhangi bir donanım üzerinde çalışması beklenir. Dolayısıyla WinCE'li tablet, telefon veya farklı geliştirme kitlerinde de bu uygulamayı hızlıca adapte etmek mümkündür.

Çizelge 5.1 Kullanılan geliştirme kartının donanımsal özellikleri

Donanımsal Özellik	Açıklamalar
Boyut	100 x 100 mm
İşlemci	400 MHz Samsung S3C2440 ARM926T
RAM	64 MB
Çakarbellek	1GB
EEPROM	256B
Harici Bellek	SD Kart Soketi
Dizisel Kapılar	3
USB	2 (1 x hizmet, 1 x aygıt)
Audio Çıkışı	var
Audio Girişi	var
Ethernet	var
LCD Arayüzü	var

Tasarlanan sistemin donanım üzerinde çalışması da bilgisayar üzerinde çalışmasına oldukça benzerdir. Sisteme güç verildikten sonra ve işletim sistemi yüklendikten sonra veri tabanının içinde bulunduğu USB arayüzlü çakar bellek takılır. Program başlatıldıktan sonra Şekil 5.6'da görüldüğü gibi kullanıcı tarafından girilen metin alınır ve kelimelere ayrılır.



Şekil 5.6 Donanım üzerinde çakar bellek ve giriş metnin gösterilmesi

Kullanıcı “Başlat” butonuna bastıktan sonra giriş metnindeki kelimeler işlenir ve TİD görüntüleri Şekil 5.7’de gösterildiği gibi ekranda oynatılır. Kit üzerindeki LCD ekran dokunmatiktir. Dolayısıyla ekrandaki klavye ile hem giriş metninin alınması hem de sistemin çıkışındaki görüntülerin gösterilmesi mümkündür.

Programın bellekte kapladığı alan 18KB civarındadır. Veri tabanında 83 kayıtlı kök kelime ve 41 adet kayıtlı sonek ve parmak alfabesi vardır. Bu durumda verilerin boyutu ise 23MB civarındadır. Kullanılan resimler JPEG formatında sıkıştırılmış olduğu için görsel verinin fazla yer kaplamadığı söylenebilir.



Şekil 5.7 Çalışan donanım üzerinde TİD görüntülerinin gösterimi

6. TESTLER VE SONUÇLAR

Önerilen sistemin yapısı ve uygulamalar önceki bölümlerde detaylı olarak incelenmiştir. Sistem tasarımı tamamlandıktan sonra, sistemin başarısının test edilmesi de önemlidir. Hedef dili işaret dili olan bir sistemin test edilmesi, hedef dili yazılıp okunabilen bir dil olan sisteme göre çok daha zordur. Eğer hedef dil yazılıp okunabilen bir dil ise, sistemin girişi ve çıkışı metin olacağı için sistemi test etmek için metot geliştirmek daha kolay olacaktır. Fakat hedef dil bir işaret dili ise, giriş ile çıkış arasında bağlantı kurmak zor olacağı için sistemin başarısını test etmek de kolay değildir.

Eğer çıkışta kullanılan görsel veri animasyon ise ve bu verinin metin tabanlı bir gösterimi varsa otomatik değerlendirme yöntemleri kullanılabilir. Bu yöntemlerde genelde cümleler arasında doğru eşleşen kelime sayısı temel alınır. Bazı yöntemlerde eşleşen kelime sayısının yanı sıra, sıralamaya da önem verilir. Test metodu her ne kadar başarılı olsa da, sonuçta karşılaştırılanlar giriş metni ile çıkışın gösterimidir, yani tam olarak çıkışta görülen görsel veri değildir. Bu yüzden tam olarak sistemin giriş ve çıkışları karşılaştırılmış sayılamaz [24]. Ayrıca sistemin başarısı tartışılırken çıkışta kullanılan görsel verinin akıcılığı, hangi oranda işaret dilini yansıttığı, gerçekçiliği gibi etmenler de göz önünde bulundurulmalıdır. Bu durumda öznel yöntem kullanmak daha iyi sonuçlar verebilir. Direkt işaret dili kullanıcıları tarafından elle değerlendirme, otomatik değerlendirmeye göre sadece çevirme performansı değil, akıcılık, görüntü kalitesi ve benzer etmenler hakkında da fikir verecektir. Fakat bu yöntemde de katılımcıların işaret dili seviyesi, işaret dilini hangi ölçüde anlayabildikleri gibi katılımcılara bağlı etmenler test yönteminin başarısı üzerinde rol oynayabilir. Bu durumda katılımcının işaret dili seviyesi yeterli değilse, katılımcının başarısı sistemin başarısını tam olarak yansıtmayabilir.

Yapılan çalışmada görsel veri olarak animasyon kullanılmamıştır. Dolayısıyla görsel verinin üretimi için ara dil olarak herhangi bir gösterime de ihtiyaç duyulmamıştır. Bu durumda otomatik değerlendirme yöntemlerinin kullanımı için görsel verinin bir gösterimine ihtiyaç duyulacaktır. Bu gösterim görsel veriden yararlanılarak elde edilebilir. Fakat sadece otomatik değerlendirme yöntemlerinin kullanılabilmesi için görsel veriden gösterim üretmek hem sonuçların gerçekçiliği açısından hem de

yapılan işin verimliliği açısından uygun olmayacaktır. Dolayısıyla yapılan çalışmada öznel değerlendirme yöntemi tercih edilmiştir.

Bu yöntemin kullanılması için TİD kullanıcılarının sistemin çevrim yeteneği hakkındaki görüşlerine başvurulmuştur. Yapılan çalışmanın değerlendirilmesi için Ankara'da bulunan Yahya Özsoy İşitme Engelliler Ortaokulu'ndan öğrenciler katılım sağlamıştır. Yapılan testlerde 8 farklı öğrenciye 20 farklı cümlenin TİD çevirisi gösterilmiş ve öğrencilerden anladıklarını öğretmenlerine anlatmaları ve tahtaya yazmaları istenmiştir. Böylece öğretmenin de onayını alarak öğrencilerin cümlenin bütününe anlayıp anlamadıklarının ve hangi aşamalarda zorluk çektiklerinin tespit edilmesi hedeflenmiştir. Testlerde kullanılan cümlelere Çizelge 6.1'de yer verilmiştir.

Çizelge 6.1 Çalışmada kullanılan metinler

	Kullanılan Cümleler
1	arkadaşınla saatkaçta buluşacaksın "mete"
2	10 dakika sonra geleceğim
3	nezaman okula gideceksin
4	2 hafta sonra okula geleceğim
5	benim yeni telefonum çok iyi
6	okul 2 ay sonra bitiyor
7	bu akşam yemek güzel
8	problem değil sonra buluşursun
9	aynı problem bende var
10	"eren" 3 gün sonra taşınıyor
11	kaç yıl aynı okul içinde kaldın
12	nezamandır "ayşe" ve sen arkadaşsınız
13	burada otur ben 5 dakika sonra geliyorum
14	nezaman yeni eve taşınacaksın
15	bu akşam "boluya" gitmek istiyorum
16	bu hafta "boluya" giden tren var mı
17	4 ay sonra benimle beraber "boluya" gelir misin
18	yarın okula kiminle beraber gideceksin
19	bu sabah okula gitmek istemiyorum
20	bu okulda eğitim güzel

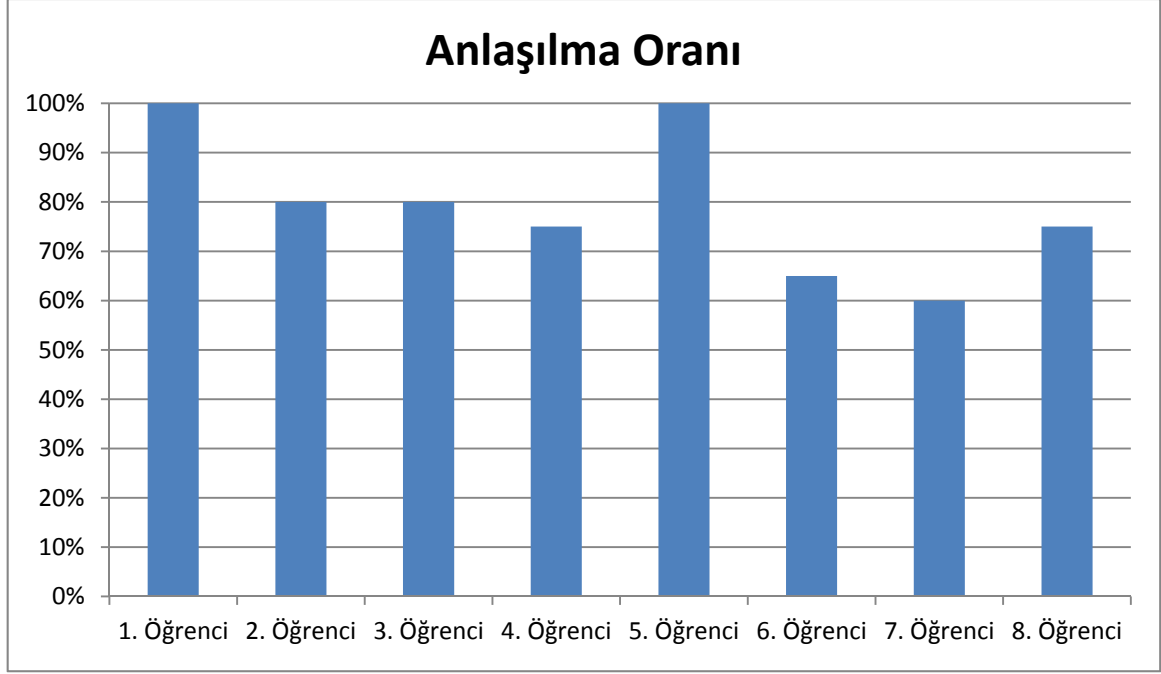
Çalışmaya yardımcı olan TİD kullanıcılarının TİD bilgi düzeyleri ve TİD kelime dağarcıkları gibi etmenler yöntemin başarısına direkt etki etmektedir. Çalışmaya katılan öğrencilerin TİD bilgisinin orta ve iyi seviyede olmasına dikkat edilmiştir. Kullanılan cümlelerde özel isim, rakam, veri tabanında olmayan kelimeler ve soru kalıpları bulunmasına özen gösterilmiştir. Böylece yapılan test ile sistemin günlük hayata uygunluğunun da test edilebilmesi hedeflenmiştir. Cümlelerin ortalama kelime sayısı 4,95'tir.

Çizelge 6.2 Testte kullanılan cümlelerin anlaşılma dağılımı (1: anladı, 0: anlamadı)

	1. öğrenci	2. öğrenci	3. öğrenci	4. öğrenci	5. öğrenci	6. öğrenci	7. öğrenci	8. öğrenci
1. cümle	1	1	1	1	1	1	0	0
2. cümle	1	1	1	1	1	1	1	1
3. cümle	1	1	1	1	1	1	1	1
4. cümle	1	1	1	1	1	1	1	1
5. cümle	1	1	0	1	1	1	0	0
6. cümle	1	1	1	1	1	1	1	1
7. cümle	1	1	1	1	1	1	1	1
8. cümle	1	0	0	0	1	0	0	1
9. cümle	1	0	1	1	1	0	0	1
10. cümle	1	1	1	1	1	1	1	1
11. cümle	1	1	0	1	1	0	1	0
12. cümle	1	1	1	1	1	1	1	1
13. cümle	1	0	1	0	1	0	0	1
14. cümle	1	1	1	0	1	1	1	0
15. cümle	1	1	1	1	1	1	1	1
16. cümle	1	1	1	1	1	1	1	1
17. cümle	1	1	0	0	1	0	1	0
18. cümle	1	0	1	0	1	0	0	1
19. cümle	1	1	1	1	1	1	0	1
20. cümle	1	1	1	1	1	0	0	1

Katılan öğrenciler bazı cümleleri anlamakta güçlük çekmişlerdir. Kullanılan cümlelerin anlaşılma dağılımı Çizelge 6.2'de verilmiştir. Bazı cümleler tamamen anlaşılabilirken, bazıları öğretmenin yardımıyla anlaşılabilir. Bazı cümlelerin, cümle içinde geçen bir kelimenin bilinmemesinden dolayı anlaşılmadığı da görülmüştür. Fakat sistemin anlaşılabilirliğine en çok etki eden konu, kullanıcının işaret dili seviyesidir. Kullanıcıların bilmedikleri kelimeler özel isim gibi parmakla heceleme yoluyla anlatıldığında sorunun ortadan kalktığı görülmüştür. Fakat işaret

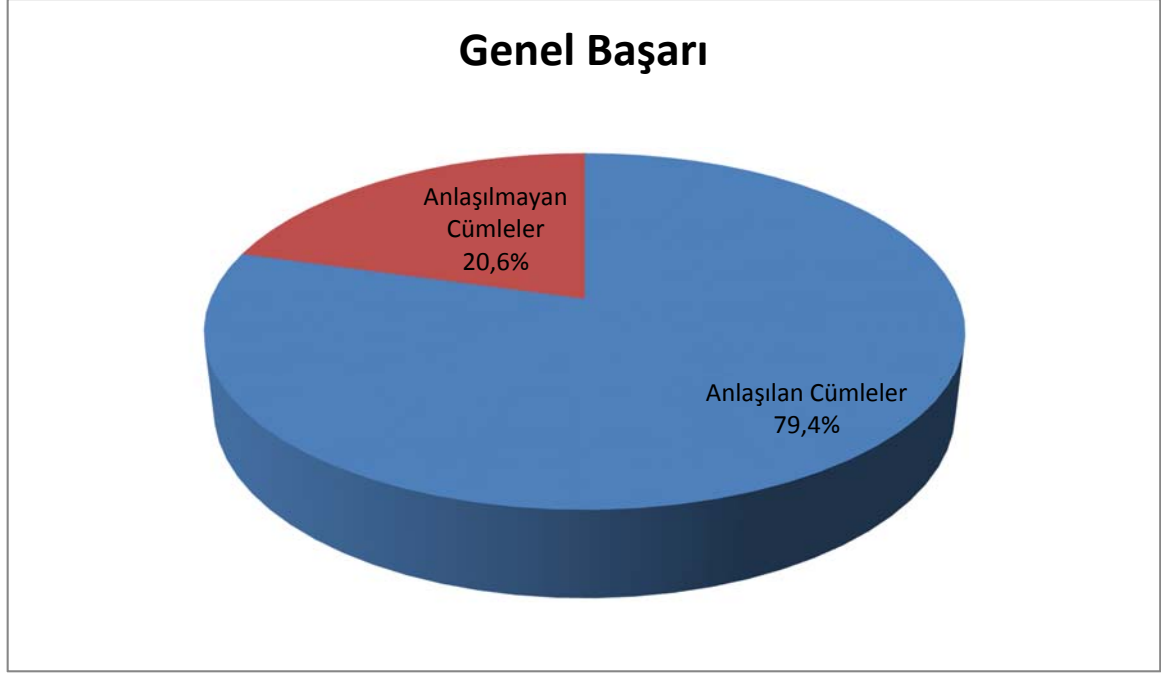
dili bilgisi az olan kullanıcıların cümleyi anlamakta güçlük çektikleri gözlemlenmiştir. Öğrencilerin hangi oranda gösterilen cümleleri anladıkları Şekil 6.1'de gösterildiği gibidir.



Şekil 6.1 Teste katılan öğrencilerin cümleleri anlama oranı

Sistemin genel başarısı öğrencilerin bireysel anlama başarılarının ortalaması olarak ölçülebilir. Bu durumda sistemin genel başarısı Şekil 6.2'de gösterildiği gibi %79,4 olarak değerlendirilebilir.

Şekil 6.3'te görüldüğü gibi en çok anlaşılan cümleler 2, 3, 4, 6, 7, 10, 12, 15 ve 16 numaralı cümleler olurken, 8, 13, 17 ve 18 numaralı cümlelerin %50 ve altında anlaşıldıkları görülmüştür.

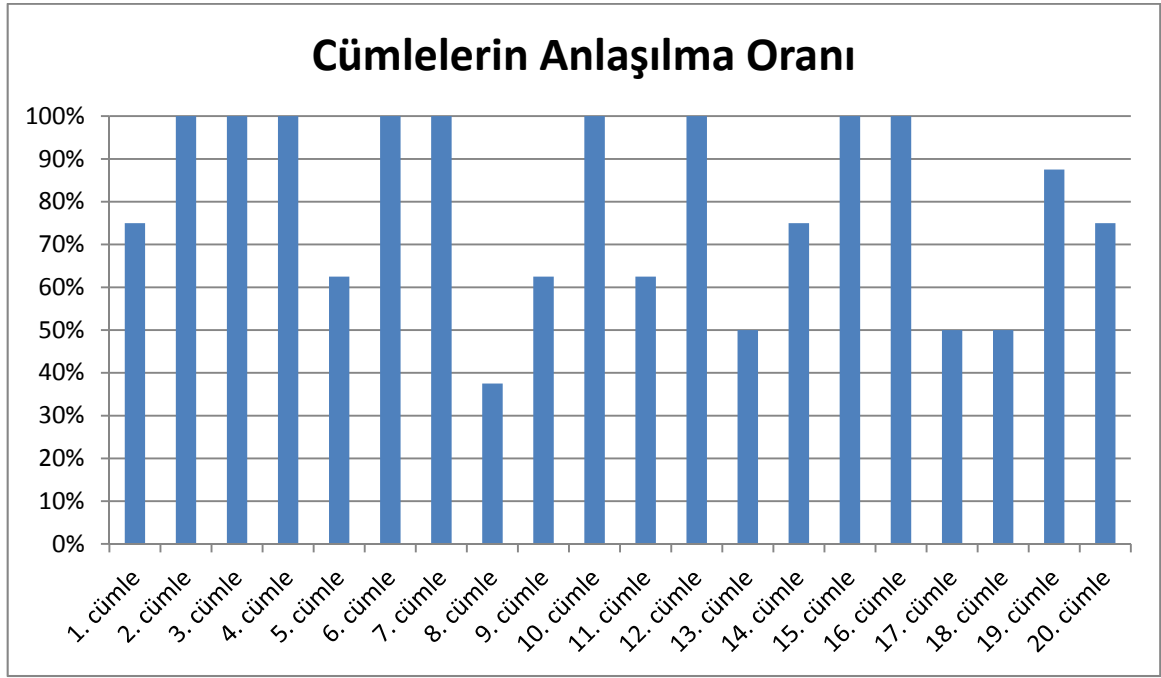


Şekil 6.2 Test sonuçları

Test sırasında dikkat çeken bir konu TİD'in Türkiye'de kullanıldığı bölgeye göre farklılıklar göstermesidir. Bu konuya Zeshan da "Aspects of Türk İşaret Dili" çalışmasında değinmiştir [1]. Örneğin bazı kelimelerin İstanbul'da ve Ankara'da kullanılan işaret dilinde farklılıklar gösterdiği bilinmektedir. Neyse ki bu kelimelerin sayısı sınırlıdır. Kelimelerin yanı sıra bazı TİD yöntemlerinin de Ankara ve İstanbul arasında farklı olması dikkat çekmektedir. Örneğin İstanbul'da kullanılan işaret dilinde zamir yüklemden önce veya sonra kullanılabilir. Fakat Ankara'da kullanılan işaret dilinde zamir sadece yüklemden önce kullanılabilir. Bu çalışmada kullanılan yöntem ve çevirilerde İstanbul'da kullanılan işaret dili temel alındığı için katılan TİD kullanıcılarının bazı konularda güçlük çektikleri görülmüştür.

Dikkat çeken bir diğer konu ise, cümledeki kelime sayısı ve cümlenin karmaşıklığı arttıkça katılımcıların çevirileri anlamakta güçlük çektikleri, bazen birden fazla kez izleme ihtiyacı duymalarıdır. Bu her ne kadar BÇ sistemlerin çoğunda görülen genel sayılabilir bir kural olsa da, özellikle işaret diline çevirme söz konusu olduğunda altı çizilmesi gereken bir konu olarak görülmelidir. Testin yapıldığı okulda edinilen bilgiye göre işaret dili kullanıcıları günlük hayatta çok karmaşık olmayan, mümkün olduğu kadar basit ve az sayıda kelime içeren cümleler kullanmaktadır. Testte kullanılan cümlelerin ortalama kelime sayısının 4,95 olduğu

düşünüldüğünde, tasarlanan sistemin günlük hayatta kullanımının uygun olduğu söylenebilir.



Şekil 6.3 Testte kullanılan cümlelerin anlaşılma oranı

Testlerde katılımcıların parmak alfabesi ile anlatılan özel isim ya da veri tabanında bulunmayan kelimeleri güçlük çekmeden anladıkları gözlemlenmiştir. Parmak alfabesi kullanımı TİD kullanıcıları arasında oldukça yaygın kullanılmaktadır. Testlerde veri tabanında olmayan kelimelerin ağırlıkta olduğu cümleler de kullanılmış ve bu cümlelerin de anlaşılır olduğu gözlemlenmiştir. Fakat bilinmeyen kelimelerin de parmak alfabesi ile anlatılması açısından da sistemin günlük hayattaki TİD kullanımı ile uyum sağladığı görülmektedir.

Testlerde sistemin akıcılık yönünden çok iyi olmadığı görülmüştür. Kelimeler arasındaki geçişlerde bazı duraksamalar görülmektedir. Bu duraksamalar çeviriyi olumsuz olarak etkilemektedir. Fakat sistem hızlandığında da bahsi geçen duraksamaların azalmasına rağmen, kelimelere karşılık gelen videoların daha hızlı gösterildiği görülmüştür. Sistemde tek zamanlayıcı birim kullanılmıştır. Bu zamanlayıcı birim hızlandığında, kelimelerin arasındaki boşluklar da azalmaktadır. Fakat bu hızlanmadan dolayı kelimeler de daha hızlı gösterilmektedir. Bu sorun iki ayrı zamanlayıcı birimi olan bir sistem tasarımı ile çözülebilir.

Kullanılan görsel verinin bellekte kapladığı alan 23MB civarındadır. Veri tabanında 83 kök kelime ve 41 sonek kayıtlı olduğu düşünülürse, bilinen tüm TİD kelimeleri veri tabanına aktarıldığında bile sistemin bellek yönünden sıkıntı çekmeyeceği tahmin edilmektedir.

7. SONUÇ ve ÖNERİLER

Türkçe metnin TİD'e dönüştürülmesi uygulamasına yönelik yazılım ve donanım tabanlı çalışma gerçekleştirildi.

Çalışmanın ilk aşamasında Visual Studio ortamında C# dili kullanılarak yazılım geliştirildi. Geliştirilen yazılım kullanıcıdan giriş olarak Türkçe metin almakta, çıkış olarak da TİD videoları vermektedir. İkinci aşamada geliştirilen yazılım donanıma aktarıldı.

Yapılan çalışma TİD kullanıcılarının katılımıyla test edildi. Testlerde katılımcıların kendilerine gösterilen TİD videolarını anlayıp anlamadıkları tespit edildi. Test sonuçlarına bakarak sistemin tatmin edici başarıya sahip olduğu gözlemlendi.

Kullanılan çevirme yöntemi ve işaret dili videoları İstanbul'da kullanılan TİD ile uyumludur. Yapılan testlerde Ankara'da kullanılan TİD'in bazı dilbilimsel kurallarının ve bazı kelimelerinin farklı olduğu görüldü. Ankara ve diğer bölgeler için de benzer bir çalışma yapılarak mevcut sisteme eklenmesi, sistemin kullanıldığı bölgeden bağımsız olarak kullanılabilmesi açısından faydalı olabilir.

Testlerde sistemin akıcılık yönünden yeterli olmadığı görüldü. Sistem yavaş çalıştığında kelimeler arasındaki geçişlerde duraksamalar olduğu, sistem hızlandığında da videoların gösteriminin gerektiğinden daha hızlı olduğu görüldü. Kelimelerin gösterim hızını belirleyen zamanlayıcı birim ile geçişlerdeki duraklamayı kontrol eden zamanlayıcı birimin birbirinden bağımsız tasarlanması bu konuda çözüm olabilir.

Geliştirilen yazılım WinCE işletim sistemi ile uyumlu olarak geliştirilmiştir. Yazılım üzerinde geliştirmeler yapılarak WinCE ve Windows işletim sistemi koşturan cep telefonu ve tablet bilgisayar gibi donanımlarda çalışması sağlanabilir. Algoritma Linux, Android gibi sık kullanılan diğer işletim sistemleri ile de uyumlu hale getirilerek çalıştırılabileceği donanım çeşitliliği artırılabilir.

Çalışma mevcut haliyle Türkçe metni TİD'E dönüştürecek şekilde tasarlanmıştır. Yapılacak ek bir çalışma ile sisteme sesten metine dönüştürme özelliği eklendiğinde, sistem sesten işaret diline çevrim yapabilir ve kullanım kolaylığı

sağlayabilir. Ek bir çalışma ile birleştirilerek işaret dilinden yazılı metne ya da sese dönüştürme özelliğinin de eklenmesi, sistemi duyma engelli kişilerle duyabilen kişiler arasında tam bir iletişim aracı haline getirebilir.

Bu çalışmada sisteme görsel veri eklemek oldukça kolaydır. Kullanıcılar sisteme video eklemek için kaynak kod üzerinde herhangi bir değişiklik yapmak zorunda kalmazlar. Doğru dizine doğru formatta görsel veri eklenmesi yeterlidir. Tasarlanan sistem bu yönüyle esnektir.

Bu tez çalışmasında tasarlanılan sistemin mevcut haliyle duyma engelli kişilerle iletişim amacıyla taşınabilir olarak kullanılabilmesi mümkündür. Yapılan çalışma ayrıca havalimanı, banka gibi halka açık alanlarda anons ya da toplu duyuruların duyma engelli kişilere ulaştırılmasında da kullanılabilir. İletişim amacıyla kullanılabileceği gibi, sistemin TİD eğitiminde de kullanılması mümkündür.

Sonuç olarak, bu tez çalışmasında geliştirilen çevirim algoritması yazılım ve donanım ortamına aktarılmış, Türkçe metni TİD'e tatmin edici bir başarı oranı ile çevirebildiği görülmüştür.

KAYNAKLAR

- [1] Zeshan, Ulrike, Aspects of Türk İşaret Dili (Turkish Sign Language), Sign Language & Linguistics, 6 (1)., pp. 43-75., ISSN 13879316, 2003.
- [2] <http://turkisaretdili.ku.edu.tr/>
- [3] Morrissey, Sara, Assistive translation technology for deaf people: translating into and animating Irish sign language, 12th International Conference on Computers Helping People with Special Needs (ICCHP 2008), 9-11 July 2008, Linz, Austria.
- [4] T. Veale, A. Conway, Cross Modal Comprehension in ZARDOS: An English to Sign-Language Translation System, Proceedings of the Seventh International Workshop on Natural Language Generation, Kennebunkport, Maine, USA, June 21-24, 1994, pp: 249-252.
- [5] I. Marshall and É. Sáfár, A prototype text to British Sign Language (BSL) translation system, Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 2, Sapporo, Japan: Association for Computational Linguistics, 2003, pp. 113-116.
- [6] J. Bungeroth and H. Ney, Statistical sign language translation, In Proc. of the Workshop on Representation and Processing of Sign Languages (LREC2004), pages 105–108, Lisbon, Portugal, 2004.
- [7] Mehrez Boulares, Mohamed Jemni, Toward an example-based machine translation from written text to ASL using virtual agent animation, International Journal of Computer Science Issues (IJCSI); Jan2012, Vol. 9 Issue 1, pp. 379-388.
- [8] Verónica López-Ludeña , Rubén San-Segundo , Carlos González Morcillo, Juan Carlos López , José M. Pardo Muñoz, Increasing adaptability of a speech into sign language translation system, Expert Systems with Applications, 2013; vol.40(4): 1312-1322.
- [9] Achraf, Othman and Mohamed, Jemni, Statistical Sign Language Machine Translation: from English written text to American Sign Language Gloss, International Journal of Computer Science Issues (IJCSI), Vol. 8, Issue 5, No 3, September 2011.
- [10] R. San-Segundo, J. M. Montero, R. Córdoba, V. Sama, F. Fernández, L. F. D'Haro, V. López-Ludeña, D. Sánchez, A. García, Design, development and field evaluation of a Spanish into sign language translation system, Pattern Analysis and Applications, May 2012, Volume 15, Issue 2, pp 203-224.

- [11] Almohimeed, Abdulaziz, Wald, M. and Damper, R.I., (2011) Arabic Text to Arabic Sign Language Translation System for the Deaf and Hearing-Impaired Community., In, EMNLP 2011: The Second Workshop on Speech and Language Processing for Assistive Technologies (SLPAT), Edinburgh, UK, Scotland, pp. 101-109.
- [12] Dimitris Kouremenosa, Stavroula-Evita Fotineab, Eleni Efthimioub and Klimis Ntalianis, A prototype Greek text to Greek Sign Language conversion system, Behaviour & Information Technology, Vol. 29, No. 5, September–October 2010, pp: 467–481.
- [13] Tirthankar Dasgupta, Sandipan Dandpat, Anupam Basu, Prototype Machine Translation System From Text-To-Indian Sign Language, Proceedings of the 13th international conference on Intelligent user interfaces - IUI '08, Gran Canaria, Spain: 2008, p. 313.
- [14] <http://www.cmpe.boun.edu.tr/tid/>
- [15] Beyza Eken, Büşra Şahin, Fatma Akbulut, Esra Akbulut, Cemil Öz, Saklı Markov Model Kullanarak Türkçe Konuşmayı ve Yazıyı İşaret Diline Çevirme, Elektrik-Elektronik ve Bilgisayar Sempozyumu 2011, 5-6-7 Ekim 2011,2011,Elazığ.
- [16] <http://www.friendlyarm.net/products/mini2440>
- [17] Dikyuva H., Zeshan U., Türk İşaret Dili - Birinci Seviye, Ishara Press, Nijmegen, 2008.
- [18] <http://www.tuik.gov.tr/>
- [19] Aslan Demir, S., Sessizliğin Dili: Türk İşaret Diline Dair Gözlemler, Bilig, Yaz 2010, Sayı 54, ss. 1-20.
- [20] <http://www.statmt.org/>
- [21] Philipp Koehn, Franz Josef Och, Daniel Marcu, Statistical Phrase-Based Translation, Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03) – Vol. 1, pp. 48-54, May-June 2003.
- [22] <http://language.worldofcomputing.net/>
- [23] Eiichiro Sumita, Example-based machine translation using DP-matching between word sequences, Proceedings of the workshop on Data-driven methods in machine translation (DMMT '01), vol. 14, pp. 1-8, 2001.
- [24] Morrissey S., Assessing three representation methods for sign language machine translation and evaluation, In: Proceedings of the 15th annual meeting of the European Association for Machine Translation (EAMT 2011), Leuven, Belgium, pp 137–144, 2011.