



**T.C.
FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ**

**Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü**

Lisans Bitirme Projesi II

TÜRKÇE EYLEM İÇEREN SESLİ KONUŞMALAR İÇİN AKIŞ ŞEMASI ÇIKARILMASI

**AHMET İHSAN SAVAŞ
1421221028**

**Bitirme Projesi Danışmanı:
PROF. DR. BURHANETTİN CAN**

İstanbul, Haziran 2020



**T.C.
FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ**

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Lisans Bitirme Projesi II

**TÜRKÇE EYLEM İÇEREN SESLİ KONUŞMALAR İÇİN AKIŞ
ŞEMASI ÇIKARILMASI**

**AHMET İHSAN SAVAŞ
1421221028**

**Bitirme Projesi Danışmanı:
PROF. DR. BURHANETTİN CAN**

Jüri Üyeleri:	İmza:
.....:
.....:
.....:
.....:
.....:
.....:

İstanbul, Haziran 2020

İÇİNDEKİLER

İÇİNDEKİLER.....	iii
ÖZET.....	v
1.GİRİŞ.....	1
Motivasyon ve Amaç.....	1
Bilim ve Mühendislik için Önemi	2
1.1 Doğal Dil İşleme	2
1.2 Doğal Dil İşlemedeki Zorluklar	3
2.TEMEL KAVRAMLAR	4
2.1 Doğal Dil İşleme Çalışma Alanları	4
2.1.1 İstatiksel Doğal Dil İşleme	4
2.1.2 Kural Tabanlı Doğal Dil İşleme	4
2.2 Doğal Dil İşleme Kullanım Alanları	4
3. SESLİ CÜMLELERİN ANLAMLI KISIMLARININ ÇIKARILMASI	5
3.1 Önerilen Yöntem	6
3.2 Kelime Yapısı İle İlgili Örnek Kurallar	7
3.3 Türkçede Sözdizimsel Ayırıştırma.....	8
3.3.1 Türkçede Sözdizimsel Ayırıştırmada Kelime Yapıları.....	9
3.3.2 Fatih Türkçe Cümle Ayırıştırıcı	10
3.3.3 Sözdizimsel Ayırıştırma ile İlgili Kurallar.....	13
3.4 Veri Seti	14
3.5 İş Planlama Sırası	15
3.5.1 İş Planlama Sırası İçin Oluşturulan Kurallar.....	16
3.5.2 Zaman Zarfı Bulunan Cümleler ile Çalışma	18
3.5.3 Zaman İfadesi İçeren Kelimeler ile Çalışma	19
3.5.4 Çıktı Sadeleştirme İşlemi	21
4.SONUÇ.....	24
5.PROJE YAPIMINDA ZORLANILAN KISIMLAR.....	26
6. İLERLEYEN DÖNEMLER YAPILABİLECEK PROJELER	27
7.KAYNAKÇA.....	28

ŞEKİL LİSTESİ

Şekil 1-Python Zemberek Çıktısı	6
Şekil 2-Türkçe bağılılık grafiği	8
Şekil 3-Türkçe Öge Ağacı [6]	10
Şekil 4-Akış Diyagramı.....	11
Şekil 5- Fatih Çözümleyici Örnek Çıktı.....	12
Şekil 6-Çözümleyici kodunun Düzenlenmiş hali	12
Şekil 7-Hatalı çıktı.....	14
Şekil 8-Örnek İş Planlama Sırası.....	15
Şekil 9-Zaman Zarfları Bulunan Cümleler.....	18
Şekil 10-Gelince Kelimesi için Örnek Çıktı.....	19
Şekil 11-Yapınca Kelimesi için Örnek Çıktı	20
Şekil 12-Düzenlenmemiş Çıktı	21
Şekil 13-Kısmen Düzenlenmiş Çıktı	22
Şekil 14-Sadeleştirilmiş Çıktı.....	23
Şekil 15-Veri Setinin Çözümleyiciye Uygulanması	24
Şekil 16-Örnek Cümle.....	25

TÜRKÇE EYLEM İÇEREN SESLİ KONUŞMALAR İÇİN AKIŞ ŞEMASI ÇIKARILMASI

ÖZET

Bu bitirme projesi raporunda Türkçede doğal dil işleme ile ilgili temel kavramlar , doğal dil işlemenin çalışma alanları ve doğal dil işlemenin kullanılan alanları açısından bilgiler vermeyi amaçlanmıştır. Doğal dil işlemenin gerekliliğini ve karşılaşılan zorluklarını ve Türkçenin dil bilgisi ile ilgili bilgiler anlatılmak istenmiştir.

Projede gelinen kısımlar ve Türkçe doğal dil işleme için geliştirilen zemberek programıyla ilgili bilgiler ve projede yapılacak kısımlara değinilmiştir. Konuşmayı yazıya aktararak “ Zemberek-NLP ” ve “ Fatih Çözümleyici “ yardımıyla Türkçe cümleleri öğelerine ayırarak örnek kural şemaları çıkarılmıştır ve zaman zarfı içeren cümlelerde de iş planlama sırası yapılarak ve iş planlama sırasının nasıl yapıldığı anlatılarak projenin temel amacı tamamlanmıştır.

1.GİRİŞ

Motivasyon ve Amaç

Öncelikle diller insanlık için büyük bir öneme sahiptir ve insanlık için iletişimin en temel ve en öncelikli halidir. Yaşamış ve yaşayan her topluluk kendini en iyi şekilde ifade edebilecek dil veya diller kullanmıştır. Diller insanların fikirlerine , düşünme biçimlerine , konuşmalarına , duygularına , yaşamlarının her alanlarına etki etmektedir. Diller insanlık için bu kadar öneme sahipken teknolojinin de geldiği konum göz önüne alındığında makinelerin ve dillerin senkronize çalışması yaşadığımız çağın en büyük isteklerinden biri olmaktadır.[1]

Projenin temel amacı en nihayetinde dilin bilgisayar ortamına aktarılmasıdır. Dili bilgisayar ortamına aktarırken cümle yapısını da bilgisayar ortamına aktarmak gereklidir ve projemizin temel amacı bu sorun olmasa da asıl amacı sesli konuşmayı yazıya aktararak cümle öğelerini komut tabanlı doğal dil işleme veya kural tabanlı doğal dil işleme yardımıyla bilgisayara aktarmaktır. Örnek olarak sesli bir şekilde aşağıda yazılan kurallara uygun bir cümle söylediğimizde bize önceden belirlemiş olduğumuz hangi kurala uygun olduğunu anlamamızı sağlayacaktır.

Projenin ikincil amacı olarak konuşmanın metine aktarılması , metinden gerekli önemli kısımlarını çıkarmaktır. Örnek olarak aşağıdaki cümle için konuşursak;

“Yerdeki çöpü al.”

Bu cümlede ki problemimiz ve çözmemiz gereken durum “yerdeki çöpü almaktır” ve bizim de eylem içeren cümlelerde çalışmamız gerekmektedir. İkinci örneğimizde ;

“Kırmızı kutuyu geç. ”

Bu cümlede de komutumuz ve cümleden istediğimiz anlamlı kısım kırmızı olan kutuyu geçmektir.

Aşağıdaki örneğimizde ;

“Kapıyı geçtikten sonra yerdeki kutuyu al.”

Bu cümlede ilk olarak “kapıyı geçmeyi “ ikinci olarak da “yerdeki kutuyu” almayı işlem sırasına koyması gerekir. Yine örnek verecek olursak

“Ali eve gelmeden önce markete git.” Cümlesi için iş planlama sırasını yazacak olursak

1. İş “Markete git”
2. İş “Eve gel” olmalıdır.

Bilim ve Mühendislik için Önemi

İlk olarak Türkçe Doğal Dil işleme ile yapılan çalışmalar az olduğu için Türkçe dili için yapılan çalışmalara bir yenisi daha eklenecektir ve projenin çıktılarını dosyaya yazdırdığımızdan dolayı ilerleyen dönemlerdeki yapılacak olan Türkçe cümle öğeleri ile ilgili projelerde kullanılabilecek veri seti olması amaçlanmıştır ve proje ile ilgili hiçbir fikri olmayan bireylere de fikir sahibi olma açısından bilgi vermektedir ve Kural Tabanlı Doğal Dil İşleme yöntemi kullanılarak Türkçe dili ile ilgili yapılacak projelerde yardımcı olması amaçlanmıştır.

1.1 Doğal Dil İşleme

Doğal dil işlemenin amacını tek cümlede anlatacak olursak doğal dil işleme bilgisayara doğrudan ses girişi ile komut vermektedir yani bilgisayarın konuşmayı anlaması denebilir .Verilen komutlar o dilin kendine has deyimler ve de mecazi anlamları da düşünülürse bilgisayarın o dilin bütün özelliklerini bilmesi olacak şekilde düşünülebilir.

Doğal dil işlemenin asıl amaçları dili çözümleme , dilin kendi anlamları ile dili anlama , yine dili kendi özellikleri ile yorumlamaktır. Doğal dil işlemenin yararlandığı alanlar ise yapay zeka , veri madenciliği , dilbilimciler , bilgisayar destekli dil bilim , bilişsel psikoloji vb.dir.[2]

Türkçe Doğal Dil işlemeyi düşünürsek Türkçe yapı olarak sondan eklemeli bir dil olup her kelimenin mecazi ,gerçek anlam ve terim anlam bakımından üçe ayrılmaktadır.

“Fazla yolcu alan gemi **battı**.”

“Garip davranışları onun gözüne **battı**.”

“Giderlerini hesaplamadığı için şirketi **batırdı**.”

1.2 Doğal Dil İşlemedeki Zorluklar

Türkçe genel olarak sondan eklemeli bir dil olduğundan dolayı sonuna gelecek eklerden dolayı istenen anlam farklı olabilir. Örneğin ; koyun ve kurbağayı ele alırsak bu isimlerin hecelerine ayırdığımızda

1-Koy-un diye ayrılırsak “koy” - fiil kökü olarak ele alındığında koy fiil anlamı olarak ortaya çıkabilir.

2-Kurbağada fiil kökü olarak “kur” sözcük öbeği fiil olarak anlamı çıkarabilir ki bu da bize morfoloji açısından çok farklı anlamlar ortaya çıkarır.

Türkçe açısından tekrar düşünürsek mecaz anlam da karşılaşılan zorluklardan biridir.

Örneğin

“Giderlerini hesaplamadığı için şirketi **batırdı**.”

Cümlesindeki “**batırdı**” kelimesi gerçek anlamının dışında kullanılmıştır ve bu da bize farklı anlam çıkaracağından karşılaşılan zorluklardan biridir.

2.TEMEL KAVRAMLAR

2.1 Doğal Dil İşleme Çalışma Alanları

2.1.1 İstatiksel Doğal Dil İşleme

İstatiksel doğal dil işleme , genellikle dilin matematiksel yapısını anlamak ve olasılık yürütmek için kullanılır. Yani kullanılan sözcükler ve cümle yapılarındaki kurallar, birer matematik formülü gibi açıklanabilmektedir. Özellikle de hesaplamalı bilimlerde uğraşan insanlar için oluşturulan bu formül vb. benzeri yapılar önem arz etmektedir. Ancak beynin karmaşık yapısında oluşturulan ve kitlelerce kabul gördüğü halde dilin kurallarıyla açıklanamayan istisnalar da hemen her dilde mevcuttur. [3]

2.1.2 Kural Tabanlı Doğal Dil İşleme

Komut tabanlı doğal dil işleme veya bir diğer ismiyle kural tabanlı doğal dil işleme , cümle içerisinde ki eylem içeren sözcükler aracılığıyla bilgisayarlara , makinelere vb. yapılara komut vererek istenen komutu yaptırmaktır. Bu sayede dil aracılığıyla insan ve bilgisayarın ortak bir payda da buluşması sağlanmaktadır ve bu da bize ilerleyen zamanlardaki teknoloji için en gerekli Doğal dil işleme çeşididir. Kural Tabanlı Doğal Dil İşleme de kurallar önceden belirlenmiş halde olmalıdır.

2.2 Doğal Dil İşleme Kullanım Alanları

Doğal dil işlemenin kullanım alanları şunlardır[4] ;

- Yazım yardımcı araçlarının geliştirilmesi
- Yazım yanlışlarının düzeltilmesi
- Bul ve değiştir

- Basılı bir metni okuma (optik olarak metin okuma) ve okuma yanlışlarını düzeltme
- Bir metnin özetini çıkarma
- Metnin içerdiği bilgiyi çıkarma
- Bilgiye erişim
- Metni anlama
- Bilgisayarla sesli etkileşim
- Bilgisayarın konuşması (metni seslendirme)
- Konuşmayı anlama (konuşmayı metne dönüştürme)
- Soru yanıt dizgeleri
- Yabancı dil okuma yardımcı araçları
- Yabancı dilde yazma yardımcı araçları
- Doğal diller arası çeviri

3. SESLİ CÜMLELERİN ANLAMLI KISIMLARININ ÇIKARILMASI

İlk olarak doğal dil işlemenin literatür araştırmasını yaparak doğal dil işleme ile ilgili teorik olarak anlamaya ve öğrenmeye çalıştım. Daha sonra ise “Zemberek” uygulamasının çalışmasını ve örnek cümleleri inceleyerek çalışma mantığını anlamaya çalışarak kendi çalışma konum olan sesli konuşmalar ile “Zemberek” programını çalışma bakımından birleştirmeye çalıştım ve veri setleri oluşturmaya çalıştım. Bu doğrultuda Sesli konuşmayı Google Api yardımıyla metin haline çevirerek birkaç basit cümle elde ettim.

Daha Sonra ise “Zemberek” uygulamasının Morfoloji modülünün yardımıyla sesli olarak verdiğim cümlelerinin kelime bazlı olarak inceledim.

Örneğin aşağıdaki cümle için incelersek(bkz. Şekil 1);

“Mehmet eve çok geç gitti.” Bu cümle için çıktımız aşağıdaki gibi olmuştur.

```
Sentence: mehmet-Noun ev-Noun çok-Adv geç-Adj git-Verb .-Punc
```

```
xxxxx
```

Şekil 1-Python Zemberek Çıktısı

Mehmet – isim

Ev (isim) + e (iyelik eki)

Çok (sıfat)

Geç (sıfat)

Git (fiil) + ti(di-li geçmiş zaman)

. (Noktalama)

3.1 Önerilen Yöntem

- **Kural Tabanlı Doğal Dil İşleme**

Komut tabanlı doğal dil işleme veya bir diğer ismiyle kural tabanlı doğal dil işleme , cümle içerisinde ki eylem içeren sözcükler aracılığıyla bilgisayarlara , makinelere vb. yapılara komut vererek istenen komutu yaptırmaktır. Bu sayede dil aracılığıyla insan ve bilgisayarın ortak bir payda da buluşması sağlanmaktadır ve bu da bize ilerleyen zamanlardaki teknoloji için en gerekli Doğal dil işleme çeşididir. Kural Tabanlı Doğal Dil İşleme de kurallar önceden belirlenmiş halde olmalıdır. Bu doğrultuda kullanacağımız yöntem olan Kural Tabanlı Doğal Dil İşleme yönteminin gereklilikleri ile projemize sesli olarak vereceğimiz cümlelerimiz için iş planlama sırasında kullanacağımız zaman zarflarını ve zaman ifadesi belirten cümlelerimizi projemizde “if” şartları ile tanımlayarak , projemizi tamamlamaya çalışacağız.

3.2 Kelime Yapısı İle İlgili Örnek Kurallar

Zemberekten elde edilen değerlerle bu değerler genel olarak kelimenin yapısı ile ilgili olarak oluşturduğum örnek kurallar aşağıdaki gibidir;

1- İSİM-FİİL

- Yağmur (isim) yağdı (fiil)

2- İSİM -İSİM -FİİL

- Annem(isim) yemek(isim) pişirdi(fiil).
- Ahmet'in(isim) halası(isim) öğretmenmiş(fiil).

3- İSİM-İSİM-İSİM-FİİL

- Yüzbaşı (İsim) Alp(İsim) buyruklar(isim) veriyor(Fiil) .
- Çocuğun (isim) yüzü (isim) gülmeye (isim) başladı (fiil).

4- SIFAT-İSİM-FİİL

- Küçük (sıfat) köpek (isim) havladı (fiil)

5- FİİL-İSİM-İSİM-FİİL

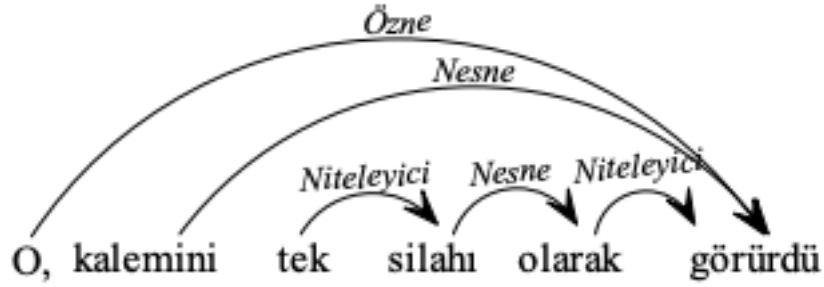
- Solmuş (fiil) çiçekler (isim) boynunu (isim) bükmüş (fiil)

6- İSİM-SIFAT-FİİL

- Akıbetiniz (isim) korkunç (sıfat) olacaktır (fiil)

3.3 Türkçede Sözdizimsel Ayırıştırma

Zemberek bize cümlede ki kelimelerin ancak yapısını doğru bir şekilde vermektedir ve bu şekilde böleceğimiz kelimeler arasında bir anlam eksikliği olacaktır çünkü Türkçe cümle yapısında fiili yapan ve fiilden etkilenen bir yapı mevcuttur ve bu bağlamı görmeden ilerlersek büyük bir yanlış olacaktır ve bu bağılılığı kelimeler arasında gösteremeyiz(bkz. Şekil2). Örneğin “büyük okuldayım” kelimelerini ele alırsak , büyük olan “okuldayım” kelimesi değil büyük olan “okul” dur.



Şekil 2-Türkçe bağılılık grafiği

3.3.1 Türkçede Sözdizimsel Ayrıştırma Kelime Yapıları

Türkçe sondan eklemeli bir dil olduğu için sözcük biçim olarak karmaşık bir yapıya sahiptir ve bu nedenle bir sözcükten türeteceğimiz kelimeler çok farklı bir anlama gelebilir.[5]

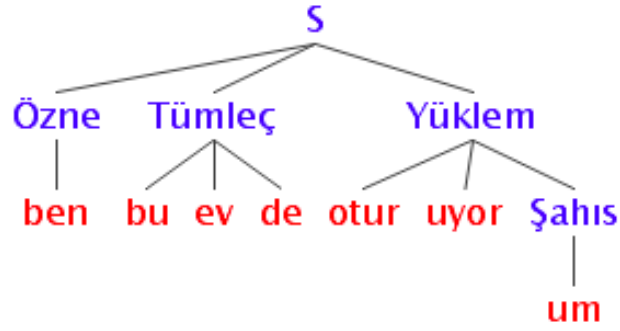
Örneğin gözlük kelimesini alırsak;

- Gözlük (isim): Görme bozukluğu olan gözlerin daha iyi görmesine veya gözleri korumaya yarayan, bir çerçeveye yerleştirilmiş çift camdan oluşan araç.
- Gözlük**çü** (isim) : Gözlük satan veya onaran kimse
- Gözlük**lü**(sıfat) : Gözlük takmış olan, gözlük kullanan
- Gözlük**süz**(sıfat) : Gözlüğü olmayan, gözlük takmamış olan
- Gözlük**çülük**(isim) : Gözlük satma işi

3.3.2 Fatih Türkçe Cümle Ayırıştırıcı

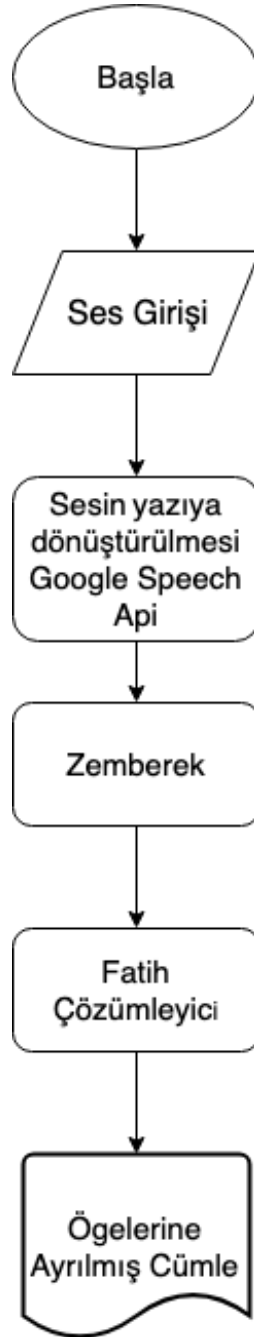
Fatih – Çözümleyici Türkçe cümleler için cümle ayrıştırması yapan açık kaynaklı bir Java kodudur. Cümleleri çözümlerken ise olasılık ile çalışır . Temel olarak Zemberek altyapısını kullanmaktadır ve projemizde bu açık kaynaklı kod üzerinden devam edeceğiz.

Fatih Çözümleyici ilk olarak Zemberek yardımıyla kelimelerin yapısına göre (isim , sıfat ,fiil vb.) ayırarak aşağıdaki şekil-3’de görüldüğü gibi cümleyi bir ağaç yapısına dönüştürür.



Şekil 3-Türkçe Öge Ağacı [6]

Fatih Çözümleyici ve Zemberek kullanarak yapmış olduğum projenin akış diyagramı şu şekildedir(bkz. Şekil 4);



Şekil 4-Akış Diyagramı

Öncelikle verdiğimiz “Ali sarı topu at.” cümlesinde fatih çözümleyiciden aldığımız çıktı aşağıdaki gibidir(bkz. Şekil 5).

Şekil 5- Fatih Çözümleyici Örnek Çıktı

```
g$
Most frequent parse is : [S[Nesne[sen][in][kalem][in][i]] [Nesne[ben]] [Yüklem[kır][dı] [Şahıs[m]]]
['--Nesne-sen', '-in', '-kalem', '-in', '-i', '', '-Nesne-ben', '', '-Yüklem-kır', '-dı', '-Şahıs-
m', '', '', '']]
--Nesne-sen -in -kalem -in -i -Nesne-ben -Yüklem-kır -dı -Şahıs-m Ahmet-MacBook-Air:fatih_py
thon ahmetihsan$
```

Şekil 6-Çözümleyici kodunun Düzenlenmiş hali

Java kaynaklı bir kod olan fatih çözümleyiciyi Java Virtual Machine yardımıyla bir python projesi haline getirerek projemize entegre etmeye çalıştım ve Veri Seti yardımıyla aşağıdaki kuralları oluşturdum.

3.3.3 Sözdizimsel Ayırıştırma ile İlgili Kurallar

Kural Tabanlı Doğal dil işleme de Sözdizimsel Ayırıştırma ile ilgili kural örnekleri ve o kurala uygun cümleler aşağıdaki gibidir;

1-) ÖZNE - NESNE – YÜKLEM

- Ali(Ö) sarı topu (N) at(Y).
- Televizyondaki konuşması(Ö) göz(N) doldurdu (Y)

2-) TÜMLEÇ – ÖZNE – YÜKLEM

- Aşığa(T) Bağdat(Ö) sorulmaz(Y).
- Köye(T) bir yabancı(Ö) gelmişti(Y).

3-) ÖZNE –TÜMLEÇ—YÜKLEM

- Mehmet(Ö) eve(T) gitti(Y).
- Adem(Ö) cennetten(T) çıkarıldı(Y).

4-) NESNE—YÜKLEM

- Akıbetini(N) biliyorsunuz(Y).

5-) ÖZNE –TÜMLEÇ—NESNE – YÜKLEM

- Komşunun tavuğu(Ö) komşuya(T) kaz(N) görünür(Y)
-

6-) TÜMLEÇ—NESNE – YÜKLEM

- Okulda(T) kimseyi(N) üzmezsiniz(Y)
-

7-) ÖZNE—ZARF—YÜKLEM

- İhtiyar adam (Ö) sabahleyin (Z) yürüyordu(Y)

3.4 Veri Seti

Projemizde Yıldız Teknik Üniversitesi tarafından yazılmış ve öğelerine ayrılmış iki bin adet cümleyi kullanmaya çalıştık ve böylece Fatih Çözümleyicinin cümle öğelerini doğru bir şekilde ayırıp veya yanlış bir şekilde ayırmadığını tek tek gözlemlemeye çalışmaya başladık ve şuan da yaklaşık yüz veya yüz elli cümle arasındayız. Projemizde bu aşamamızda şöyle bir problem ile karşılaştık kullanmış ve düzenlemiş olduğumuz Fatih Çözümleyici uzun ve karışık cümlelerde her zaman doğru sonucu verememektedir.

Örneğin “ Kıbrıs bir Avrupa Birliği üyesidir ” cümlesi için konuşursak veri setimizde bu cümlenin ayrımı şöyle olmuştur; “Kıbrıs(Özne) bir Avrupa Birliği(Belirtisiz Nesne) üyesidir(Yüklem) ”. Fakat bizim Çözümleyiciden aldığımız çıktımız ise aşağıdaki Şekil 7 ‘deki gibi olmuştur.

```
Derivation 0 "kıbrıs/prop" "bir/num" "avrupa/prop" "birlik/noun i/acc" "üye/noun si/p3s dir/aux"
Derivation 1 "kıbrıs/prop" "bir/num" "avrupa/prop" "birlik/noun i/p3s" "üye/noun si/p3s dir/aux"
Derivation 2 "kıbrıs/prop" "bir/num" "avrupa/prop" "bir/num liğ/+lÜk i/acc" "üye/noun si/p3s dir/aux"
Derivation 3 "kıbrıs/prop" "bir/num" "avrupa/prop" "bir/num liğ/+lÜk i/p3s" "üye/noun si/p3s dir/aux"
Derivation 4 "kıbrıs/prop" "bir/adj" "avrupa/prop" "birlik/noun i/acc" "üye/noun si/p3s dir/aux"
Derivation 5 "kıbrıs/prop" "bir/adj" "avrupa/prop" "birlik/noun i/p3s" "üye/noun si/p3s dir/aux"
Derivation 6 "kıbrıs/prop" "bir/adj" "avrupa/prop" "bir/num liğ/+lÜk i/acc" "üye/noun si/p3s dir/aux"
Derivation 7 "kıbrıs/prop" "bir/adj" "avrupa/prop" "bir/num liğ/+lÜk i/p3s" "üye/noun si/p3s dir/aux"
Traceback (most recent call last):
  File "fatih_parser22.py", line 75, in <module>
    res = max(set(maxilist), key = maxilist.count)
ValueError: max() arg is an empty sequence
```

Şekil 7-Hatalı çıktı

Ancak bizim çalışacağımız kısım basit fiil cümleleri olacağından şu aşamada çözümleyicinin bizim istediğimiz doğrultuda olduğunu düşünmekteyiz.

3.5 İş Planlama Sırası

Projemizin bu kısmında birden fazla fiil içeren cümleler için cümle içerisindeki eylemleri sıralamaya çalışmaktayız. Örneğin ”Kapıyı geçtikten sonra yerdeki kutuyu al.” Bu cümlede ilk olarak “kapıyı geçmeyi “ ikinci olarak da “yerdeki kutuyu” almayı işlem sırasına koyması gerekir. Buradaki kilit noktamız cümle içerisindeki “önce” ,”sonra” ,”ardından “ gibi zaman zarfı içeren kelimelerdir. Bu yüzden cümle içerisinde öncelikle bu kelimelerin olması bizim için öncelikli kısımdır.

Örneğin “Ali eve gel sonra markete git.” cümlesi için çıktı Şekil 8’de verilmiştir. Bu cümle için iş sıramız;

- 1- Eve Gel
- 2- Markete git

```
['Ali eve gel ', ' markete git']
-----
Derivation 0  "ali/prop" "ev/noun e/dat" "gel/verb"
      Parse 0 [S[Özne[ali]][Tümleç[ev][e]][Yüklem[gel]]]
Derivation 1  "ali/adj" "ev/noun e/dat" "gel/verb"
      Parse 0 [S[Tümleç[ali][ev][e]][Yüklem[gel]]]
      Parse 1 [S[Özne[ali]][Tümleç[ev][e]][Yüklem[gel]]]
Most frequent parse is : [S[Özne[ali]][Tümleç[ev][e]][Yüklem[gel]]]
['--Özne-ali', '', '-Tümleç-ev', '-e', '', '-Yüklem-gel', '', '', '']
--Özne-ali -Tümleç-ev -e -Yüklem-gel -----ikinci kısım-----
-----
Derivation 0  "market/noun e/dat" "git/verb"
      Parse 0 [S[Tümleç[market][e]][Yüklem[git]]]
Most frequent parse is : [S[Tümleç[market][e]][Yüklem[git]]]
['--Tümleç-market', '-e', '', '-Yüklem-git', '', '', '']
--Tümleç-market -e -Yüklem-git 1 . yapılacak is ['--Özne-ali', '', '-Tümleç-
ev', '-e', '', '-Yüklem-gel', '', '', '']
2 . yapılacak is ['--Tümleç-market', '-e', '', '-Yüklem-git', '', '', '']
Ahmet-MacBook-Air:fatih_python ahmetihsan$
```

Şekil 8-Örnek İş Planlama Sırası

3.5.1 İş Planlama Sırası İçin Oluşturulan Kurallar

Projemizde iş planlama şeması oluşturmak için kurallar oluşturmamız gereklidir çünkü cümleleri belirli bir kurala göre bölerek daha sonra da sıralamaktayız. Örnek verecek olursak ;

“Ali okula gitmeden önce kırtasiyeye git.” Cümlesi için yazacağımız kod parçası cümlede ki “ önce ” sözcüğünü algılayıp daha sonra bu sözcüğe göre oluşturulan kural doğrultusunda cümlenin iş sırasını çıkartabilmelidir. Asıl odak noktamız kelimelerdir. Verdiğim örnek doğrultusunda çıkardığım kurallar şunlardır;

1-“Önce” kelimesi için çıkarmış olduğum kural:

1.cümle +”önce”+”2.cümle”

“Önce” kelimesi için yapılacak iş sırası:

İlk olarak 2. Cümle daha sonra ise 1. Cümledir.

Örnek;

“Ali okula gitmeden önce kırtasiyeye git”

1.yapılacak eylem “kırtasiyeye git”

2.yapılacak eylem “okula git”

2-“Sonra” kelimesi için çıkarmış olduğum kural:

1.cümle+”sonra”+”2.cümle”

“Sonra” kelimesi için yapılacak iş sırası:

İlk olarak 1. Cümle daha sonra ise 2. Cümledir.

Örnek;

“Ayşe eve geldikten sonra markete git.”

1.yapılacak eylem “eve gel”

2.yapılacak eylem “markete git”

3-“Ardından” kelimesi için çıkarmış olduğum kural:

1.cümle+”ardından”+”2.cümle”

“Ardından” kelimesi için yapılacak iş sırası:

İlk olarak 1. Cümle daha sonra ise 2. Cümledir.

Örnek;

“Veli markete git ardından eve gel.”

1.yapılacak eylem “markete git”

2.yapılacak eylem “eve gel”

4-“Sonunda” kelimesi için çıkarmış olduğum kural:

1.cümle+”sonunda”+”2.cümle”

“Sonunda” kelimesi için yapılacak iş sırası:

İlk olarak 1. Cümle daha sonra ise 2. Cümledir.

Örnek;

“Elif dışarda çok oynama sonunda markete gideceksin.”

1.yapılacak eylem “dışarda çok oynamamak”

2.yapılacak eylem “markete gitmek”

5-“Zaman” kelimesi için çıkarmış olduğum kural:

1.cümle+”zaman”+”2.cümle”

“Zaman” kelimesi için yapılacak iş sırası:

İlk olarak 1. Cümle daha sonra ise 2. Cümledir.

Örnek;

“Zeynep okuldan geldiğin zaman yemek ye.”

1.yapılacak eylem “okuldan gelmek”

2.yapılacak eylem “yemek ye”

6-“En son” yapısı için çıkarmış olduğum kural:

1.cümle+”sonunda”+”2.cümle”

“En son” yapısı için yapılacak iş sırası:

İlk olarak 1. Cümle daha sonra ise 2. Cümledir.

Örnek;

“Elif eve gel en son markete gideceksin.”

- 1.yapılacak eylem “eve gel”
- 2.yapılacak eylem “markete gitmek”

3.5.2 Zaman Zarfı Bulunan Cümleler ile Çalışma

Projemizin bu kısmında önce , sonra ve ardından vb. gibi zaman zarfları bulunan cümlelerden oluşmuş metin dosyasında yazmış olduğumuz kodun başarısı üzerinde çalıştım. Bu kısımda asıl amaç yazmış olduğumuz kodun hangi tür bütünleşik cümleleri daha iyi bir şekilde cümle öğelerine ayırıp daha sonra da iş sırasına koymasıydı. Örneğin en çok başarı sağlanan cümleler genellikle içerisinde tanımlı özne bulunduran cümlelerde daha iyi başarı oranı sağlanmaktadır. Bu sonuçla Şekil 9-da ki gibi bir çıktı elde ettim.

```
[ '--Özne-hasan', '', '-Tümleç-iş', '-e', '', '-Yüklem-git', '', '', '' ]
[ '--Nesne-kal', '+An-an', '', '-iş', '-ler', '-i', '', '-Yüklem-yap', '', '' ]
[ '--Tümleç-market', '-e', '', '-Yüklem-git', '-ti', '-Şahıs-ler', '', '', '' ]
[ '--Yüklem-çocuk', '-Şahıs-lar', '', '', '-Tümleç-ev', '-e', '', '-Tümleç-g
me', '', '-den', '', '', '' ]
[ '--Özne-ayşe', '', '-Tümleç-market', '-e', '', '-Yüklem-git', '', '', '' ]
[ '--Tümleç-fır', '-ın', '-dan', '', '-Nesne-ekmek', '', '-Yüklem-al', '', '' ]
[ '--Nesne-fiş', '-i', '-ni', '', '-Yüklem-çek', '-neg-me', '', '', '' ]
[ '--Nesne-ceviz', '-ler', '-i', '', '-Yüklem-kır', '', '', '' ]
[ '--Nesne-araba', '-yı', '', '-Yüklem-hazırla', '', '', '' ]
-----
103 adet cümleden öğelerine ayrılan sayısı 68
Ahmet-MacBook-Air:fatih_python ahmetihsan$
```

Şekil 9-Zaman Zarfları Bulunan Cümleler

3.5.3 Zaman İfadesi İçeren Kelimeler ile Çalışma

Projemizde cümlelerin iş planlama sırası yapmak için kullandığı zaman zarflarının yanı sıra birkaç kelime de kullanmış oldum. Bu kısımda amaç projemizin iş planlama sırası yapmak için sadece zaman zarflarına bağlı kalmaması ile açıklanabilir.

Örneğin ; “Mehmet eve gelince ödevlerini yap.” Cümlesinde bir zaman zarfı bulunmadığı için projemiz burada devreye giremeyecekti ancak “gelince” kelimesini de diğer zaman zarfları gibi kullanırsak bu problemi ortadan kaldıracaktır. Öyleyse “gelince” için yazmış olacağımız kural aşağıdaki gibi olacaktır.

“Gelince” kelimesi için çıkarmış olduğum kural:

1.cümle+”gelince”+”2.cümle”

Yukarıdaki örnek için iş sırası;

İlk olarak 1. Cümle daha sonra ise 2. Cümledir.

Örnek;

“Mehmet eve gelince ödevlerini yap”

1.yapılacak eylem “eve gel”

2.yapılacak eylem “ödevlerini yap”

“Gelince” kelimesi için örnek çıktımız , Şekil 10 ‘da ki gibi olmuştur.

```
Bir şey söyleyin :
Söylediğiniz şey : Ali eve gelince ödevlerini yap
ali
eve
gelince
ödevlerini
yap
kullanılan kelime : gelince
['Ali eve gel', ' ödevlerini yap']
-----
En çok geçen çözüm : [S[Özne[ali]][Tümleç[ev][e]][Yüklem[gel]]]
['--Özne-ali', '', '-Tümleç-ev', '-e', '', '-Yüklem-gel', '', '', '']
--Özne-ali -Tümleç-ev -e -Yüklem-gel -----
---ikinci kısım---
En çok geçen çözüm : [S[Nesne[ödev]][leri][ni]][Yüklem[yap]]]
['--Nesne-ödev', '-leri', '-ni', '', '-Yüklem-yap', '', '', '']
--Nesne-ödev -leri -ni -Yüklem-yap -----
Cümlelerin is sırası
1 . yapılacak is ['--Özne-ali', '-Tümleç-ev', '-e', '-Yüklem-gel']
2 . yapılacak is ['--Nesne-ödev', '-leri', '-ni', '-Yüklem-yap']
```

Şekil 10-Gelince Kelimesi için Örnek Çıktı

Bir diğ er zaman ifadesi iç eren kelimemiz ise “yapınca”. Bu kelime iç in ö rnek cümlemiz ise “Veysel ö devlerini yapınca bilgisayar oyna.”

“Yapınca” kelimesi iç in ç ıkarmış olduğ um kural:

1.cümle+”yapınca”+”2.cümle”

Yukarıdaki ö rnek iç in iş sırası;

İlk olarak 1. Cümle daha sonra ise 2. Cümledir.

1.yapılacak eylem “ö devlerini yap”

2.yapılacak eylem “bilgisayar oyna”

“Yapınca” kelimesi iç in ö rnek ç ıktımız , Ş ekil 11 ‘de ki gibi olmuştur.

```
Bir ş ey söyleyin :
Söylediğ iniz ş ey : Ayşe ö devlerini yapınca top oyna
ayşe
ö devlerini
yapınca
top
oyna
kullanılan kelime : yapınca
['Ayşe ö devlerini yap', ' top oyna']
-----
En ç ok geç en ç özü m : [S[Özne[ayşe]][Nesne[ö dev][leri][ni]][Yüklem[yap]]]
['--Özne-ayşe', '', '-Nesne-ö dev', '-leri', '-ni', '', '-Yüklem-yap', '', '', ''
]
--Özne-ayşe -Nesne-ö dev -leri -ni -Yüklem-yap -----
---ikinci kısım---
En ç ok geç en ç özü m : [S[Özne[top]][Yüklem[oyna]]]
['--Özne-top', '', '-Yüklem-oyna', '', '', ''
]
--Özne-top -Yüklem-oyna -----
Cümle nin is sırası
1 . yapılacak is ['--Özne-ayşe', '-Nesne-ö dev', '-leri', '-ni', '-Yüklem-yap']
2 . yapılacak is ['--Özne-top', '-Yüklem-oyna']
```

Ş ekil 11-Yapınca Kelimesi iç in Ö rnek Ç ıktı

3.5.4 Çıktı Sadeleştirme İşlemi

Projemizde iş planlama sırasını ve iş planlama işlemlerinin nasıl yapıldığından bahsettikten sonra en son yapılacak işlemde proje çıktımızı düzenlemek olmuştur. Bunun nedeni ise proje çıktımızın ve proje çıktımızdan elde ettiğimiz verileri biz anlayabiliriz ancak herkes anlayamayabilir bu yüzden çıktıyı kısmen biraz daha sadeleştirerek projeye nokta koymaya çalıştım.

Şekil 12 'de görmüş olduğumuz düzenlenmemiş çıktı da birden çok fazla ve dışarıdan bakıldığı zaman anlaşılması biraz zor olmaktadır.

```
['Ali eve gel ', ' markete git']
-----
Derivation 0  "ali/prop" "ev/noun e/dat" "gel/verb"
      Parse 0 [S[Özne[ali]][Tümleç[ev][e]][Yüklem[gel]]]
Derivation 1  "ali/adj" "ev/noun e/dat" "gel/verb"
      Parse 0 [S[Tümleç[ali][ev][e]][Yüklem[gel]]]
      Parse 1 [S[Özne[ali]][Tümleç[ev][e]][Yüklem[gel]]]
Most frequent parse is : [S[Özne[ali]][Tümleç[ev][e]][Yüklem[gel]]]
['--Özne-ali', '', '-Tümleç-ev', '-e', '', '-Yüklem-gel', '', '', '']
--Özne-ali -Tümleç-ev -e -Yüklem-gel -----ikinci kısım-----
-----
Derivation 0  "market/noun e/dat" "git/verb"
      Parse 0 [S[Tümleç[market][e]][Yüklem[git]]]
Most frequent parse is : [S[Tümleç[market][e]][Yüklem[git]]]
['--Tümleç-market', '-e', '', '-Yüklem-git', '', '', '']
--Tümleç-market -e -Yüklem-git 1 . yapılacak is ['--Özne-ali', '', '-Tümleç-
ev', '-e', '', '-Yüklem-gel', '', '', '']
2 . yapılacak is ['--Tümleç-market', '-e', '', '-Yüklem-git', '', '', '']
Ahmet-MacBook-Air:fatih_python ahmetihsan$
```

Şekil 12-Düzenlenmemiş Çıktı

Şekil 13’ te ki çıktımızda ise önceki çıktımıza göre nispeten biraz daha anlaşılabilir ve Biraz daha sonuç odaklı olmuştur.

```
Bir şey söyleyin :
Söylediğiniz şey : Ayşe Eve gelmeden önce markete git
ayşe
eve
gelmeden
önce
markete
git
kullanılan bağlaç : önce
['Ayşe eve gel ', ' markete git']
-----
En çok geçen çözüm : [S[Tümleç[market][e]][Yüklem[git]]]
['--Tümleç-market', '-e', '', '-Yüklem-git', '', '', '']
--Tümleç-market -e -Yüklem-git -----
---ikinci kısım---
En çok geçen çözüm : [S[Özne[ayşe]][Tümleç[ev][e]][Yüklem[gel]]]
['--Özne-ayşe', '', '-Tümleç-ev', '-e', '', '-Yüklem-gel', '', '', '']
--Özne-ayşe -Tümleç-ev -e -Yüklem-gel -----
Cümlelerin is sırası
1 . yapılacak is ['--Tümleç-market', '-e', '-Yüklem-git']
2 . yapılacak is ['--Özne-ayşe', '-Tümleç-ev', '-e', '-Yüklem-gel']
(base) Ahmet-MacBook-Air:fatih_python ahmetihsan$
```

Şekil 13-Kısmen Düzenlenmiş Çıktı

Daha sonra ise elde ettiğimiz bu çıktıları bir dosyada tutarak ilerleyen dönemlerde yapılacak olan bu tarz projelerde bir fikir vermesi açısından projemizin çıktılarını dosyada tutmaya karar verdik ve projemizin çıktılarını bir dosyaya yazdırdım.

```
Ayşe eve gel sonra markete gir
kullanılan bağlaç : sonra
Cümlenin is sırası
1.is
  Özne ayşe
  Tümleç ev
  Yüklem gel
2.is
  Tümleç market
  Yüklem gir
```

Şekil 14-Sadeleştirilmiş Çıktı

Şekil 14 ‘deki çıktıyı önceki çıktılarımızla karşılaştırsak , öncelikle elde ettiğimiz çıktılardan daha yalın ve daha sade olmuştur bu sayede ilerleyen dönemlerdeki yapılacak olan Türkçe cümle öğeleri ile ilgili projelerde kullanılacak veri seti olması amaçlanmıştır ve proje ile ilgili hiçbir fikri olmayan bireylere de fikir sahibi olma açısından bilgi vermektedir ve Kural Tabanlı Doğal Dil İşleme yöntemi kullanılarak Türkçe dili ile ilgili yapılacak projelerde yardımcı olması amaçlanmıştır.

4.SONUÇ

İlk olarak Türkçe Doğal Dil işleme ile yapılan çalışmalar az olduğu için zorlandık fakat daha sonra ise Türkçe cümle öğelerine ayırma işlemi için hali hazırda çalıştırılacak uygulama sayısı yok denecek kadar az olduğu için “Fatih Çözümleyici” kullanarak ayırdığımız cümleleri farklı yazılımlar ile test etme imkanı bulamadık ancak kullanmış olduğumuz çözümleyiciye bir takım kurallar ekleyerek gerçekleştirmek istediğimiz kısımları gerçekleştirdik ve çözümleyicinin doğruluğunu test etmek için Yıldız Teknik Üniversitesi tarafından oluşturulmuş olan 2000 öğelerine ayrılmış cümle ile karşılaştırma yapma gereği hissettik.

```
[ '--Özne-sibel', '', '-Tümleç-kırmızı', '-ışık', '-ta', '', '-Yüklem-al', '' ]  
[ '--Özne-pınar', '', '-Nesne-ip', '', '-Yüklem-al', '', '', '' ]  
69 adet cümleden öğelerine ayrılan sayısı 44
```

Şekil 15-Veri Setinin Çözümleyiciye Uygulanması

“ Fatih Çözümleyici” ‘ ye vermiş olduğumuz 69 adet cümleden 44 adeti cümle öğelerine ayrılabilmiştir.(bkz. Şekil 15)

Bu doğrultuda elde edilen çıktı kullanmış olduğumuz çözümleyiciyi ilerleyen dönemlerde kullanmamız için yeterli oldu ve projenin ikinci kısmına geçerek cümledeki iş planlama sırası kısmına geçtik ve projede iş planlama sırası için kural yazarak çözümleyiciyi geliştirdik ve bu sayede birden fazla fiil içeren cümleler için çıktılar elde etmiş olduk.

Örneğin aşağıdaki cümleyi incelersek

“Kapıyı geç sonra yerdeki kutuyu al.”

Bu cümlede ilk olarak “kapıyı geçmeyi “ ikinci olarak da “yerdeki kutuyu” almayı işlem sırasına koyması gerekir.

Genel olarak en son yapmış olduğumuz kısımlarla birlikte büyük oranda iş sırasını yapmaya çalıştık. Buradaki asıl kısım en nihayetinde cümle yapısıdır. Cümle yapısında belirli bir özne ve nesne ve bir yüklem bulunduğunda cümle öğelerine ve de iş sırasına daha iyi ayrılabilir. fakat bazı cümlelerde özne olmasa dahi belirli yüklem ve nesne bulunduğunda da Çözümleyicimiz öğelerine ayırabilmektedir.

Örnek olarak ;

“Eve gel sonra markete git.” Cümlesini ele alırsak bu cümlede belirli bir özne bulunmamasına rağmen çözümleyicimiz bu cümle için iş sırasını çıkarmış ve cümle öğelerini doğru bir şekilde bölmüştür. Şekil 16 'da gösterilmiştir.

```
['Eve gel ', ' markete git']
-----
Derivation 0  "ev/noun e/dat" "gel/verb"
  Parse 0 [S[Tümleç[ev][e]][Yüklem[gel]]]
Most frequent parse is : [S[Tümleç[ev][e]][Yüklem[gel]]]
['--Tümleç-ev', '-e', '', '-Yüklem-gel', '', '', '']
--Tümleç-ev -e -Yüklem-gel  -----ikinci kısım-----
-----
Derivation 0  "market/noun e/dat" "git/verb"
  Parse 0 [S[Tümleç[market][e]][Yüklem[git]]]
Most frequent parse is : [S[Tümleç[market][e]][Yüklem[git]]]
['--Tümleç-market', '-e', '', '-Yüklem-git', '', '', '']
--Tümleç-market -e -Yüklem-git  1 . yapılacak is ['--Tümleç-ev', '-e', '',
Yüklem-gel', '', '', '']
2 . yapılacak is ['--Tümleç-market', '-e', '', '-Yüklem-git', '', '', '']
Ahmet-MacBook-Air:fatih.python ahmetihcan$
```

Şekil 16-Örnek Cümle

Elde ettiğimiz genel verileri bir tabloda göstermiş olursak. (Tablo-1)

	Toplam Cümle Sayısı	Öğelerine Ayrılan Cümle Sayısı	%Yüzde
Zaman Zarfı içeren Cümleler	103	68	%66
Zaman Zarfı içermeyen Cümleler	69	44	%63

Tablo 1 İstatistiksel Kıyaslama

Böyle bir tablo(Tablo -1) karşımıza çıkar , Önce ve Sonra kelimeleri bulunan Zaman Zarfı içeren cümlelerin oranının fazla olmasının nedeni daha önce de anlattığım gibi belirli özne ve yüklem olmasıdır. Eğer bu doğrultudaki cümlelerimizi çözümleyiciye verirsek bu oranların artması muhtemeldir.

5.PROJE YAPIMINDA ZORLANILAN KISIMLAR

İlk olarak Türkçe Doğal Dil işleme ile yapılan çalışmalar az olduğu için zorlanılmıştır. Daha sonra ise Türkçe cümle öğelerine ayırma işlemi için hali hazırda çalıştırılacak uygulama sayısı yok denecek kadar az olduğu için “Fatih Çözümleyici” kullanarak ayırdığımız cümleleri farklı yazılımlar ile test etme imkanı bulamadık ve bu sorun bizim için en önemli zorlanılan kısım olmuştur.

6. İLERLEYEN DÖNEMLER YAPILABİLECEK PROJELER

Türkçe cümle öğeleri ile ilgili projelerde kullanılabilecek veri seti oluşturarak ilerleyen dönemlerde yapılabilecek Kural Tabanlı Türkçe Doğal Dil İşleme yöntemi ile temel düzeyde veri seti bulma açısından yardımcı olması amaçlanmıştır ve proje ile ilgili hiçbir fikri olmayan bireylere de fikir sahibi olma açısından bilgi vermektedir ve eklenecek çok kapsamlı kurallar ile Türkçe Sesli Asistan ve Sesli komut ile komut verebileceğimiz herhangi bir aygıt ve robotlar vb. projelere temel düzeyde yardımcı olabilir.

7.KAYNAKÇA

- [1] Prof. Dr. Eşref Adalı , “Türkçe Belgelerden Bilgi Çıkarımı İçin Bir Tümüleşik Mimari”, Makale, İTÜ Fen Bilimleri 2009
- [2]Dr. Gülşen Eryiğit, 2006. Türkçe’nin Bağlılık Ayrıştırması, Doktora Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [3]Bozşahin, H. C. and Oflazer, K., 1994. Turkish Natural Language Processing Initiative: An Overview, In Proceedings of Third Turkish Symposium On Artificial Intelligence, Middle East Technical University, Ankara, Turkey.
- [4]Sadi Evren Şeker , Doğal Dil İşleme 2015
- [5]Kemal Oflazer – Türkçe Doğal Dil işleme
- [6] Fatih Çözümleyici - A Generic Syntactic Parser For Turkic Languages By Harun Reşit ZAFER
- Doğal Dil İşleme Rehberi-<https://medium.com/türkiye/doğal-dil-işleme-rehberi-2c4c41260f74>
- Zemberek Türkçe Doğal Dil İşleme Eklentisi, <http://zembereknlp.blogspot.com>
- Yıldız Teknik Üniversitesi 2000 öğelerine ayrılmış cümle - <http://kemik.yildiz.edu.tr>
- Google Speech Api