

**Doğal Dil İşleme Teknikleri Kullanılarak
Türkçe Mobil Asistan
Yazılımı Geliştirilmesi**

YÜKSEK LİSANS TEZİ

Gökhan ÇELİKKAYA

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Programı

Mayıs 2015

**Doğal Dil İşleme Teknikleri Kullanılarak
Türkçe Mobil Asistan
Yazılımı Geliştirilmesi**

YÜKSEK LİSANS TEZİ

**Gökhan ÇELİKKAYA
(504111517)**

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Yrd. Doç Dr. Gülşen ERYİĞİT

Mayıs 2015

İTÜ, Fen Bilimleri Enstitüsü'nün 504111517 numaralı Yüksek Lisans Öğrencisi **Gökhan ÇELİKKAYA**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**Doğal Dil İşleme Teknikleri Kullanılarak Türkçe Mobil Asistan Yazılımı Geliştirilmesi**” başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Yrd. Doç Dr. Gülşen ERYİĞİT**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Doç.Dr. Şule Öğüdücü**
İstanbul Teknik Üniversitesi

Yrd.Doç.Dr. Arzucan Özgür
Boğaziçi Üniversitesi

.....

Teslim Tarihi : **04 Mayıs 2015**
Savunma Tarihi : **26 Mayıs 2015**

ÖNSÖZ

Bu tez 0073.STZ.2013-1 numaralı SAN-TEZ projesi kapsamında Sanayi Bakanlığı ve Huawei firması tarafından desteklenmiştir. Başta tez boyunca bana yardımcı olan danışmanım Yrd. Doç. Dr. Gülşen Eryiğit ve Huawei firmasına teşekkür ederim.

Mayıs 2015

Gökhan ÇELİKKAYA

İÇİNDEKİLER

Sayfa

ÖNSÖZ	v
İÇİNDEKİLER	vii
KISALTMALAR.....	ix
ÇİZELGE LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xiii
ÖZET	xv
SUMMARY	xvii
1. GİRİŞ	1
1.1 Tezin Amacı.....	1
1.2 Tezin Araştırma Niteliği	2
1.2.1 Veri oluşturma	2
1.2.2 Karar verme algoritması	3
1.2.3 Parametre çıkarımı.....	3
1.3 Literatür Araştırması	3
1.4 Olası Senaryolar	5
2. SİSTEM TASARIMI	7
2.1 Sunucu - İstemci Arası Haberleşme	8
3. MOBİL UYGULAMA	13
3.1 Uygulama Arayüzü.....	13
3.2 Uygulamanın Çalışma Akışı.....	13
3.3 Desteklenen Operasyonlar.....	18
3.3.1 Sesli arama.....	18
3.3.2 Mesaj gönderme	20
3.3.3 E-Posta gönderme.....	20
3.3.4 Uygulama başlatma	21
3.3.5 İnternette arama	21
3.3.6 Haritada arama.....	21
3.3.7 Trafik bilgisi	21
3.3.8 Hava durumu görüntüleme	22
3.3.9 Döviz bilgisi	22
4. SORGUNUN ANLAŞILMASI ve SUNUCU YAPISI.....	25
4.1 Sorgu Ön İşleme	25
4.2 DDİ Araçları.....	27
4.2.1 Varlık ismi tanıma.....	27
4.2.2 Bağlılık ayrıştırıcı	30

4.3 Sorgu Sınıflandırma.....	31
4.3.1 Kural tabanlı yöntemler	32
4.3.2 İstatistiksel yöntemler.....	34
4.4 Parametre Çıkarımı.....	35
4.4.1 Arama	35
4.4.2 Mesaj	36
4.4.3 E-Posta.....	37
4.4.4 Harita	37
4.4.5 Uygulama	37
4.4.6 İnternette arama	37
4.4.7 Hava durumu	37
4.4.8 Döviz	38
5. TEST SONUÇLARI	45
5.1 Veri	45
5.2 Bağlılık Ayrıştırıcı Başarısı	46
5.3 Sınıflandırma Başarısı	47
5.3.1 Verinin işlenmesi	48
5.3.2 Kullanılan algoritmalar.....	48
5.3.3 Sınıflandırma testleri	49
5.4 Parametre Çıkarımı Başarısı.....	54
5.4.1 Verinin hazırlanması.....	54
5.4.2 Parametre çıkarımı testleri.....	54
6. SONUÇ VE ÖNERİLER	57
KAYNAKLAR.....	59
EKLER	63
EK A.1	65
ÖZGEÇMİŞ	69

KISALTMALAR

DDİ	: Doğal Dil İşleme
NLP	: Natural Language Processing
NER	: Varlık İsmi Tanıma (Named Entity Recognizer)
POS	: Part of Speech
HTTP	: Hypertext Transfer Protocol
REST	: Representational State Transfer
UTC	: Coordinated Universal Time
SOAP	: Simple Object Access protocol
JSON	: JavaScript Object Notation
API	: Application Programming Interface
CoNNL	: Conference on Natural Language Learning
Regex	: Regular Expression
SVM	: Support Vector Machine
HMM	: Hidden Markov Model
CRF	: Conditional Random Field
MEMM	: Maximum Entropy Markov Model
IMST	: ITUMetuSabancıTreebank
IOS	: iPhone Operating System

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 2.1 :Sunucudan Dönen Cevabın Açıklaması.....	11
Çizelge 3.1 :Gönderilen İsteğin Parametreleri	17
Çizelge 4.1 : "Antalya’da hava nasıl" cümlesi için bağıllık analizi örneği	31
Çizelge 4.2 :Sınıf bazında çıkartılan parametreler.	35
Çizelge 5.1 :Veri Bilgisi	45
Çizelge 5.2 :Veri Tipi Tanımı	46
Çizelge 5.3 :IMST Verisi ile Eğitilmiş Bağıllık Ayrıştırıcı Modelinin Üretilen Veri Üzerindeki Başarımı.....	46
Çizelge 5.4 :Üretilen Veriye Nokta Eklendikten Sonra IMST Verisi ile Eğitilmiş Bağıllık Ayrıştırıcı Modelinin Başarısı	47
Çizelge 5.5 :Üretilen Verinin 10-kat Çapraz Doğrulama Sonucu Ortalama Başarısı.....	47
Çizelge 5.6 :Gövdeleme ve NER adımlarından Önce ve Sonra Sınıflandırma Başarısı.....	50
Çizelge 5.7 :Bayes Net Algoritmasının Sınıf Bazında Başarımı	50
Çizelge 5.8 :K-En Yakın Komşu Algoritmasının Sınıf Bazında Başarımı	51
Çizelge 5.9 :0.25 Güvenirlik Faktörlü Karar Ağacının Sınıf Bazında Başarımı.....	51
Çizelge 5.10 :0.50 Güvenirlik Faktörlü Karar Ağacının Sınıf Bazında Başarımı.....	51
Çizelge 5.11 :1 Güvenirlik Faktörlü Karar Ağacının Sınıf Bazında Başarımı.....	52
Çizelge 5.12 :Lojistik Regresyon Algoritmasının Sınıf Bazında Başarımı	52
Çizelge 5.13 :Naive Bayes Algoritmasının Sınıf Bazında Başarımı.....	52
Çizelge 5.14 :Naive Bayes Multinomial Algoritmasının Sınıf Bazında Başarımı....	53
Çizelge 5.15 :SVM Algoritmasının Sınıf Bazında Başarımı	53
Çizelge 5.16 :Test Edilen Uygulamaların Sınıf Bazında Başarısı.....	55

ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 2.1 : Sistem Çalışma Prensibi	8
Şekil 3.1 : Mobil Uygulama Arayüzü	14
Şekil 3.2 : Mobil Uygulama Ses Dinleme	14
Şekil 3.3 : Mobil Uygulama Çalışma Akışı	15
Şekil 3.4 : Sesten Metne Dönüşüm Seçim Prosedürü	16
Şekil 3.5 : Sesli Arama Akışı	19
Şekil 3.6 : Uygun E-Posta Uygulaması Seçme Ekranı	20
Şekil 3.7 : İnternette Arama İşlemi Ekran Görüntüsü	21
Şekil 3.8 : Harita İşlemi Ekran Görüntüsü	22
Şekil 3.9 : Mobil Uygulama Ekran Görüntüleri	23
Şekil 4.1 : Sunucu Tarafındaki Sistem Akışı	26
Şekil 4.2 : Varlık İsmi Tanıma Aracı ile İşaretleme Akışı	29
Şekil 4.3 : Varlık İsmi Tanıma Aracı Eğitiminde Kullanılan Veri Örneği	29
Şekil 4.4 : Sınıflandırma Akışı	33
Şekil 4.5 : Arama Servisi Parametre Çıkarım Akışı	36
Şekil 4.6 : Harita Servisi Parametre Çıkarım Akışı	39
Şekil 4.7 : Uygulama Servisi Parametre Çıkarım Akışı	40
Şekil 4.8 : İnternette Arama Servisi Parametre Çıkarım Akışı	41
Şekil 4.9 : Hava Durumu Parametre Çıkarım Akışı	42
Şekil 4.10 : Döviz Servisi Parametre Çıkarım Akışı	43
Şekil 5.1 : Sınıflandırma Verisinin Bir Kısımını	48

**Doğal Dil İşleme Teknikleri Kullanılarak
Türkçe Mobil Asistan
Yazılımı Geliştirilmesi**

ÖZET

Günümüzde tüketiciler hem istedikleri işlemleri yerine getirmek hem de ihtiyaç duydukları bilgi ve içeriğe erişmek için mobil cihazlarıyla basit ve kolay etkileşim kurmanın yollarını aramaktadırlar. Bu doğrultuda mobil cihazlarla sesli olarak etkileşim kurmaya imkan sağlayan kişisel asistan uygulamaları gittikçe popüler hale gelmektedir. Özellikle Apple'ın SIRI uygulamasının yaygınlaşmasından sonra, çeşitli firmalar benzer ürünler geliştirmişlerdir. Fakat çıkan ürünler çoğunlukla İngilizceyi ve Amerika Birleşik Devletleri, İngiltere gibi İngilizce konuşulan ülkelerde kullanılabilen servisleri desteklemektedirler. Türkçe için, 2015 yılı Nisan ayındaki güncelleme ile Türkçe desteği kazanan SIRI haricinde gelişmiş doğal dil işleme yöntemleri kullanan, birçok servisi esnek bir yapıda destekleyen, kullanıcının isteğini doğru anlama açısından başarıyı yüksek bir mobil kişisel asistan uygulaması henüz geliştirilememiştir. Bu tezin amacı gelişmiş doğal dil işleme yöntemlerinin bu uygulamalar üzerindeki etkisini göstermek ve akıllı telefon ve tabletler üzerinde Türkçe doğal dil girdisi ile çalışacak yüksek başarımlı bir kişisel asistan uygulaması geliştirmektir.

Bir sanal asistan, içerisinde yapay zekanın hemen her tür bileşenini bulunduran bir uygulamadır. Bunlar arasında, ses tanıma, ses üretimi, algılanan sesin anlamsal olarak işlenmesine yönelik doğal dil işleme (DDİ) yöntemleri ve hatta 3 boyutlu hologramların kullanımı ile sistemin bir insanı taklit etmesi (görüntü tanıma, görüntü oluşturma vb.) sayılabilir. **Mobil sanal asistanlar** ise şimdilik sadece kullanıcı ile ses üzerinden iletişim kuran ve sistemden beklenen istekleri yerine getirip sorgulara sesli cevap döndüren uygulamalardır. Bu tez kapsamında geliştirilen uygulama, sesli yanıt döndürme özelliği bulunmayan bir mobil sanal asistan uygulamasıdır. Son 15 yıl içerisinde, Türkçe için başarılı sayılabilecek ses tanıma, ses üretimi, doğal dil işleme bileşenleri geliştirilmiştir ve bu konudaki araştırmalar halen aktif olarak devam etmektedir. Ancak bu ayrı bileşenler henüz başarılı üst düzey uygulamalarda bir araya getirilebilmiş değildir. Bu tez çerçevesinde, **Türkçe** için geliştirilen bu çalışmaların (sesi metine dönüştürme, sözcük çözümleyicileri, cümle çözümleyicileri, varlık ismi saptayıcıları, metin normalleştiricileri vb.) öncelikle “kişisel asistan uygulamaları” konusuna yönelik alan adaptasyonları yapılmış ve sonrasında bileşenler ürüne yönelik ortak bir çatı altında birleştirilmiştir. Yapılan ön incelemelerde, Türkçe için az sayıdaki mevcut sanal asistan uygulamasının, temel olarak kural tabanlı veya basit doğal dil işleme bileşenlerine sahip oldukları veya bu bileşenleri hiç içermeyip sadece görselleştirmeye yöneldikleri görülmüştür. Bu nedenle, dili anlama ve istenilen işlemleri yerine getirme konusundaki başarımlarının oldukça düşük olduğu gözlemlenmektedir. Son yıllarda Türkçe DDİ üzerine geliştirilen yeni yöntemlerin bu uygulamalar için bir araya getirilmesi ile çok daha yüksek başarımlı kişisel mobil

asistan uygulamaları oluşturulabileceği tarafımızdan öngörülmüştür. Bu çalışma ile geliştirilen uygulamanın bu alanda öncü bir rol oynaması hedeflenmektedir.

Tez kapsamında geliştirilen uygulamanın aşağıdaki işlemler ve servisler ile ilgili sesli komutları anlayıp yerine getirmektedir.

- Sms gönderimi (*Gökhan'a yarın akşam gelemeyeceğimi mesaj at*),
- E-posta gönderimi (*Gökhan'a yarın akşam gelemeyeceğimi mail at*),
- Numarayı veya kayıtlı kişiyi arama (*Gökhan'ı ara*),
- Telefonda yüklü olan uygulamaları çalıştırma (*Facebook'u başlat*),
- İnternet üzerinden arama başlatma (*İstanbul'u Google'da ara*),
- Hava durumu bilgisini alma (*Bugün hava nasıl olacak?*),
- Harita servisi (*Beşiktaş'a nasıl giderim?*),
- Döviz çevirme işlemleri (*10 dolar kaç lira*),
- Trafik bilgisini alma (*Şu an İstanbul'da trafik nasıl?*).

Geliştirilen sistem 2 kısımdan oluşmaktadır.

- **Mobil uygulama**, sesi yazıya dönüştürme, sunucuya istek gönderme ve gelen yanıtı gösterme veya gerçekleştirme görevlerinden sorumludur.
- **Sunucu**, yazılı olarak gelen isteği DDİ araçlarını kullanarak anlamak ve gerekiyorsa 3. parti diğer servislerden bilgi toplayarak yanıt döndürmekten sorumludur.

Gelen sorgu sunucu tarafında işlenmeden önce olası dil bilgisi hatalarını azaltmak için metin normalleştiriciden geçirilir. Ardından DDİ araçlarının sonuçlarının da kullanıldığı kural ve istatistiksel yöntemlerin birleşiminden oluşan melez bir yapı ile sorgunun hangi servis ile ilgili olduğu anlaşılır. Geliştirilen melez model, test verimiz üzerinde %98.30 başarı göstermiştir. Ardından belirlenen servis için anlamlı olan parametreler çıkartılır ve gerekli olan servisler için 3. parti kaynaklardan bilgi çekilerek yanıt döndürülür. Tüm sistemin başarısı ortalama %70.79 olarak belirlenmiştir.

Development of a Turkish Mobile Assistant Software Using Natural Language Processing Techniques

SUMMARY

Since the interaction with mobile devices increased enormously in recent years, user friendly interfaces facilitating human-machine interaction became very important. Since natural language is the most natural way of communication for humans, mobile assistant applications became very popular. Especially after SIRI gained popularity, several companies developed similar applications and released to the application markets. Nevertheless most of these applications supports English and provide services mostly for English-speaking countries. Unfortunately, the challenges caused by the typological differences of some natural languages also appears in mobile applications and the generated systems which are mostly on English, could not be adapted easily to other languages such as Turkish.

In the current state, there is no application for Turkish that uses advanced natural language processing (NLP) solutions, supports many services in a flexible structure and has a high success rate on understanding users' request except SIRI which started to support Turkish language after IOS 8.3 update in April 2015. Even though SIRI began to support Turkish, its success rate and supported service count is very low compared to its English version. The goal of this thesis is to illustrate the impact of using natural language solutions on mobile assistant applications and develop a new application that will support Turkish natural language input with high success rate. The application developed performs operations on mobile device (e.g. calling someone) and retrieve and show information (e.g. weather forecast information).

A virtual assistant is an application that contains almost all components of artificial intelligence. Voice recognition, voice generation, natural language processing components to understand the voice semantically, and even usage of 3 dimensional holograms to simulate humans (image recognition or generation) can be considered in this aspect. **Mobile assistants** are applications that interacts with user over voice processing and understands user's intention and accomplish his/her request.

In the last 15 years successful voice recognition, voice generation and natural language processing components have been developed for Turkish. The researches in these fields are still continuing actively. However the components have not been integrated in a high level application. Within this thesis, domain adaptation is applied to these researches (speech to text, morphological analysis, dependency parser, named entity recognizer, text normalizer) in personal mobile assistant domain and combined in an application.

It is observed that Turkish mobile assistant applications which there is only few, developed in a rule based approach and does not use or uses very basic natural language

processing components. Therefore, their success rate on understanding and executing user's query is very low. For example for a query such as "Ankara'da hava bugün nasıl? (How is the weather today in Ankara)", such applications return the weather forecast of the current location (e.g. "Istanbul") instead of the requested location ("Ankara"). Implementing Turkish NLP researches to the software would increase the success of these applications. This work is the first academic study focusing on the language related problem of a Turkish Mobile Assistant and shows the state-of-the art in this field. It is expected that the application and study in this thesis will become a pioneer in its field of practice.

The application developed is able to understand and fulfill voice queries on following domains.

- Sending message (*Gökhan'a yarın akşam gelemeyeceğimi mesaj at*) (*Send message to Gökhan that I will not come tomorrow evening*),
- Sending e-mail (*Gökhan'a yarın akşam gelemeyeceğimi mail at*) (*Send email to Gökhan that I will not come tomorrow evening*),
- Calling a number or contact (*Gökhan'ı ara*) (*Call Gökhan*),
- Launching applications on the device (*Facebook'u başlat*) (*Launch Facebook*),
- Search something on the Internet¹ (*İstanbul'u Google'da ara*) (*Search Istanbul on Google*),
- Retrieve weather information² (*Bugün hava nasıl olacak?*) (*How will be the weather today*),
- Map Direction³ (*Beşiktaş'a nasıl giderim?*) (*How can I go to Besiktas*),
- Currency exchange⁴ (*10 dolar kaç lira*) (*How many liras is 10 dollars*),
- Traffic information⁵ (*Şu an İstanbul'da trafik nasıl?*) (*How is traffic in Istanbul*).

Developed system consists of 2 modules which interacts with each other using defined JSON protocol.

- **Mobile application** is responsible for 1. capturing the spoken request, 2. converting it to written text 3. transferring the user requests to the server side and 4. displaying and/or performing the requested operation according to the result received from the server.
- **Server service** is responsible for processing the user's query and understand its true intention. It also collects information from third party web services if needed, and finally, compiles the results and sends it back to the client.

¹Google search

²World Weather Online

³Google Maps

⁴Google Currency Service

⁵Yandex Traffic Service

Google's speech recognition API for Android⁶ is used in mobile application in order to capture the user spoken queries and convert them into written messages. The speech recognizer returns more than one text suggestions for a given entry and each suggestion has its own confidence level. If any of the produced suggestions has more than this threshold value, it is assumed to be the correct suggestion. Otherwise, the application presents a list of text suggestions to the user in. After this stage, the selected query text is sent to the server along with the user's context, which includes the user id, current location and time. The returned answer from the server is then either executed on the phone or displayed to the user.

On the server-side incoming user query is first processed with NLP tools, and then the query is mapped to one of the supported operations through a hybrid approach of rule-based and statistical classification. Different algorithms have been tested in statistical classification step and best result is obtained with support vector machine (SVM) algorithm with **95.80% classification accuracy**. Total classification success of the hybrid approach is measured as **%98.30**. After this stage, the required parameters for each operation are extracted from the query using the assigned analysis by the automatic natural language processing modules. Whole system success is measured as **70.79%**.

⁶<http://developer.android.com/reference/android/speech/package-summary.html>

1. GİRİŞ

Günümüzde insanlar, akıllı telefon ve tablet gibi mobil cihazlarını daha kolay ve basit kullanarak işlerini daha hızlı halletmek ve bilgiye erişmek istemektedir. Bu bağlamda ses-tabanlı mobil asistan uygulamaları son zamanlarda çok popüler olmuştur. SIRI [1] uygulamasının büyük bir başarı elde etmesi sonrasında birçok benzer uygulama geliştirilmiş ve marketlere eklenmiştir. Maalesef geliştirilen uygulamaların çoğu İngilizceyi desteklemekte ve bu dili konuşan ülkelerde servis sağlamaktadır. Tez kapsamında, bu ihtiyaçtan yola çıkılmış ve Türkçe için çalışacak yüksek başarılı bir uygulama geliştirilmiştir.

1.1 Tezin Amacı

Her geçen gün daha da çok büyüyen iletişim dünyasında akıllı mobil telefon ve tablet bilgisayarların önemi artık tartışılmaz bir boyuta ulaşmıştır. Kullanıcılar, haberleşme, eğlence, hava durumuna bakma ve bilgi edinme gibi pek çok gündelik işini bu mobil cihazları kullanarak yapmaktadır. Apple ve Google gibi büyük bilişim şirketleri, mobil telefon ve tablet kullanıcılarının bu işlemleri daha kolay yerine getirebilmesi için İngilizce doğal dil ile etkileşebilecek akıllı mobil asistan uygulamaları geliştirmiştir. Bu firmaların yanı sıra birçok küçük ve orta ölçekli firma da benzer yazılımlar geliştirmiştir. Kullanım kolaylığı sağlamanın ötesinde bu uygulamaların, kişisel üretkenliği artırmak, trafikte güvenli araç kullanılmasını sağlamak ve fiziksel engeli bulunan kişilerin de mobil cihazları kullanabilmesine yardımcı olmak gibi çeşitli faydaları bulunmaktadır.

Bu tezin amacı akıllı telefon ve tabletler üzerinde Türkçe doğal dil girdisi ile çalışacak bir kişisel asistan uygulaması geliştirmektir. Uygulama, kullanıcının mobil cihaz üzerinde gerçekleştirilebilecek işlemlerle ilgili isteklerini yerine getirebilecek ve hava durumu, trafik bilgileri ve döviz fiyatları gibi belli başlı servislerle ilgili sorgularına yanıt verebilecektir.

Günümüzde, Siri [1] ve Google Now [2] gibi İngilizce doğal dil girdisi ile çalışan, kullanıcı isteklerini yerine getirebilme ve çok çeşitli sorguları yanıtlayabilme anlamında başarımı yüksek mobil asistan uygulamaları varken, Türkçe için doğal dilde etkileşebilecek benzer başarımda bir uygulama bulunmamaktadır. IOS 8.3 güncellemesi ile birlikte Siri Türkçe desteği sunmuştur fakat yine de Türkçe için desteklediği servis sayısı ve kalitesi İngilizce için olana kıyasla çok düşüktür. İngilizce için geliştirilen uygulamalar, kullanıcıların söylediklerini anlayıp onların isteklerini yerine getirebilmek için, sesten yazıya dönüşüm, doğal dil işleme, yapay zeka, anlamsal web sistemleri ve dağıtık hesaplama gibi pek çok teknoloji ve bilgisayar bilimleri alanından faydalanmaktadır. Türkçenin yapısal zorlukları (sondan eklemeli bir dil olması gibi), doğal dil işleme için gerekli veri kaynaklarının azlığı (anlamsal sözlük ve işaretli veri gibi), diğer dillerdeki doğal dil işleme çalışmalarının Türkçe'ye uyarlanamayışı gibi sebeplerden dolayı endüstride Türkçe doğal dil girdisi ile çalışan benzer uygulamaların azlığı ve yetersizliği dikkat çekmektedir.

Tez kapsamında, Türkçe doğal dil işleme konusunda akademik alanda yapılan çalışmalardan ve araçlardan faydalanılmıştır. Resmi metin verileri için derlenmiş bu araçlar ihtiyaç doğrultusunda uygulamaya yönelik olarak uyarlanmıştır. Kullanıcının niyetini anlamak ve bu niyeti doğru şekilde yerine getirebilmek için algoritmalar geliştirilmiştir.

1.2 Tezin Araştırma Niteliği

Son yıllarda Türkçe için doğal dil işleme alanında pek çok çalışma yapılmıştır. Bu tez kapsamında önceki çalışmalarla ortaya konan alt seviye doğal dil araştırma çıktıları üst düzey bir DDİ uygulaması geliştirilmek üzere uyarlanmış ve bir araya getirilmişlerdir. Ayrıca doğal dil işleme bileşenlerini aktif olarak kullanmanın mobil asistan uygulamalarının başarısı üzerindeki etkisi gösterilmiştir. Bu kapsamda yapılan araştırma faaliyetleri aşağıda açıklanmıştır.

1.2.1 Veri oluşturma

Tez kapsamında kullanılan Türkçe doğal dil işleme araçları resmi metin üzerinde eğitildiğinden, tez alanındaki başarısı düşük çıkmıştır. Bu araçları uygun veri ile

eğitmek başarımı artıracağı öngörüldüğünden bu kapsamda 620 cümlelik cümle analizi yapılmış ve 750 cümlelik varlık isimleri işaretlenmiş veri kümeleri hazırlanmıştır. Ayrıca gelen sorgunun sınıflandırması adımı istatistiksel yöntemde kullanmak üzere 1000 cümlelik veri işaretlenmiştir. Uygulamanın nihai başarısını artırmak ve ölçmek için de 578 cümleden oluşan test verisi hazırlanmıştır.

1.2.2 Karar verme algoritması

Projemizin en önemli kısımlarından birisini oluşturacak karar verme algoritması, gelen sorgunun desteklenen hangi servis ile ilgili olduğunu belirlemektedir. Örneğin, "Yarın hava nasıl" sorgusu "hava durumu" servisine, "Gökhan'a akşam toplantıya gelemeyeceğimi mesaj at" sorgusu mesaj servisine yönlendirilmektedir. Karar verme algoritması, kural ve istatistik tabanlı yöntemlerin ve bir arada kullanıldığı melez bir yapıda geliştirilmiştir. Ayrıca doğal dil işleme araçlarının kullanımının bu modülün başarısı üzerindeki etkisi de gösterilmiştir.

1.2.3 Parametre çıkarımı

Uygun servis belirlendikten sonra sorgudan ilgili parametreler çıkartılmıştır. Örnek olarak "Gökhan'a akşam toplantıya gelemeyeceğimi mesaj at" sorgusundan, mesaj servisine iletilmek üzere "Gökhan" alıcı kısmı, "akşam toplantıya gelemeyeceğim" ise içerik parametresi olarak çıkartılmıştır. Bu parametreleri çıkartmak için doğal dil işleme araçlarının çıktıları aktif olarak kullanan algoritmalar geliştirilmiştir. Bu araçları kullanmanın başarıya olan etkisi benzer diğer uygulamalar ile kıyaslama şeklinde yapılmıştır.

1.3 Literatür Araştırması

Mobil kişisel asistan uygulamaları, son yıllarda gittikçe yaygınlaşan ve kullanıcıların mobil telefon ve tablet bilgisayarlar ile doğal dilde etkileşimini sağlayan uygulamalardır. Bu uygulamalar, girilen sorguyu çeşitli doğal dil işleme yöntemleri kullanarak anlamayı ve kullanıcıya, isteğine uygun yanıt döndürmeyi amaçlamaktadır.

Bu alanda İngilizce için yapılmış birçok çalışma ve uygulama bulunmaktadır, ardı ardına yayınlar [3] yapılmakta, patenler [4] alınmaktadır. Apple'ın geliştirdiği

Siri ve Google'ın Android işletim sistemli cihazlar için ürettiği Google Now uygulamaları en yaygın kullanılan kişisel asistan uygulamalarıdır. Telefon işlemlerini çalıştırmak (arama, mesaj gönderme, uygulama başlatma, vb.), genel bilgi sorularını cevaplamak (Dünya ile Ay arasındaki uzaklık ne kadar?), sohbet etmek, önceden tanımlanmış özelleştirilmiş servislerden bilgi çekmek (hava durumu, borsa, vb.) gibi eylemler bir arada çalışmaktadır. Bu uygulamalar gelen sorguyu, doğal dil işleme yöntemleri kullanarak ilgili servise yönlendirmekte ve ardından istenen işlemi gerçekleştirmektedir.

Türkçe için de son yıllarda doğal dil işleme alanında alt seviyede önemli araştırmalar yapılmış, başarılı biçimbilimsel analiz [5], varlık ismi tanıma [5] [6] ve bağıllık ayrıştırıcı [5], [7] araçları geliştirilmiştir.

Türkçe için de son yıllarda mobil kişisel asistan alanında geliştirilmiş uygulamalar bulunmaktadır. Fakat bu çalışmalar İngilizce için yapılanlara kıyasla desteklediği servis sayısı, doğal dil işleme teknolojilerinin kullanımı ve başarı oranı açısından çok yetersizdir. Bunlardan en yaygın kullanılanı CEYD-A [8] uygulamasıdır. Bunun dışında Turkcell Mobil Asistan [9] ve Asistan B [10] uygulamaları da Türkçe için geliştirilmiş yaygın olarak kullanılan uygulamalardandır. Turkcell Mobil Asistan uygulaması hava durumu, döviz bilgisi, arama yapma gibi servisleri desteklese de, daha çok firmanın kendi servislerinin sesli sorgulanmasına yönelik kural tabanlı bir yapıda geliştirilmiştir. Asistan B [10] uygulaması da genel olarak kural tabanlı yöntemler ile geliştirilmiş olup gerekli doğal dil işleme bileşenlerini kullanmamaktadır. Bu çalışmalara göre daha amatör bir çabayla ortaya çıkmış İDRİS [11] yazılımı ise çok fazla problem ile ortaya çıkmış sonrasında da uygulama marketlerinden çekilmiştir. CEYD-A uygulaması her ne kadar DDİ araçlarını kullanmasa da daha gelişmiş bir kural yapısına sahiptir ve basit bir öğrenme yöntemiyle bu kurallarını da geliştirebilmektedir. Uygulama kullanıcılarının kendi ihtiyaçları doğrultusunda kural tanımlamasına da izin vermektedir. Türkçe için geliştirilen bu uygulamalar kural tabanlı bir yöntem izlendiğinden rastlanan temel problem, kullanıcıların sorguları sadece belli bir şablon ile sormak zorunda kalmasıdır. Örnek olarak hava tahmini bilgisi "İzmir hava durumu" şeklinde sorulmalıdır, "Hava yarın İzmir'de nasıl olur acaba" sorusuna doğru yanıt dönmemektedir.

Türkçe için geliştirilen uygulamaların dışında Nisan 2015'te yayınlanan IOS 8.3 güncellemesi ile Siri uygulamasına Türkçe desteği gelmiştir. Yapılan incelemeler sonucunda Siri'nin kullanıcı arayüzü açısından oldukça iyi seviyede olduğu ve Türkçe doğal dil işleme bileşenlerini de aktif olarak kullandığı saptanmıştır. Bu sayede diğer Türkçe uygulamalarına kıyasla daha başarılı yanıtlar döndürdüğü gözlenmiştir.

1.4 Olası Senaryolar

Proje kapsamında geliştirilen uygulama aşağıdaki işlemler ve servisler ile ilgili sesli komutları anlayıp yerine getirmektedir.

- Arama Yapma
 - Gökhan'ı ara
 - 0507 257 08 25'i arar mısın
- Mail Gönderme
 - Gökhan'a mail at adres bilgilerine internet sitemizden erişebilirsiniz
 - gokhancelikkaya@gmail.com'a ben gelmeden toplantıya başlamayın e-postasını gönder
- Uygulama Başlatma
 - Geleriği başlat
 - Okey uygulamasını çalıştır
- Hava Durumu Sorgulama
 - Yarın Ankara'da hava nasıl olacak
 - Bugün hava nasıl
- Döviz Kuru Sorgulama
 - 25 euro kaç lira
 - 5 sterlin kaç dolar eder

- Harita Servisler
 - Kadıköy'e nasıl giderim
 - Kadıköy'deki penguen kitabevi nerede
- İnternette Arama
 - İstanbul'u Google'da ara
 - Atatürk'ü webde ara
- Trafik Sorgulama
 - Şu an İstanbul'da trafik nasıl
 - Trafik nasıl

2. SİSTEM TASARIMI

Bu kısımda geliştirilen sistemin tasarımı anlatılmıştır. Sistem iki kısımdan oluşmaktadır; mobil uygulama ve sunucu servisi. Projenin çalışma prensibi Şekil 2.1’de gösterilmiştir ve şu şekildedir.

1. **Kullanıcının sesli olarak uygulamaya komut vermesi:** Bu aşamada kullanıcı gerçekleşmesini istediği cümleyi söyler ve bu cümle konuşmadan metne çevrilir. Örnek olarak "Gökhan’a bu akşam gelemeyeceğimi mesaj at" cümlesi verilebilir. Konuşmadan metne çevirmek için Google’ın Android işletim sistemi için sağladığı servis [12] kullanılmıştır. Bu servisin Türkçe doğal dil için doğruluğu test edilmiştir ve bir çok uygulama konuşmayı metne çevirmek için bu servisi kullanmaktadır.
2. **Sunucuya istek gönderimi:** Projede yoğun işlem gücü gerektiren aşamalar olduğundan, bu işlemleri mobil cihaz üzerinde yapmak gerek işlemci gücü, gerek hafıza kapasitesi, gerek de pil kullanımı açısından uygun değildir. Bu nedenlerden dolayı sunucu kullanılmasına karar verilmiştir. Bu aşamada mobil uygulama, kullanıcının sorgusunu düz metin olarak sunucuya yollayarak, cevap bekleme aşamasına geçmektedir. Mobil uygulama ile sunucunun haberleşmesi REST web servis mimarisi üzerinden yapılmıştır. Projede mobil uygulama sesi yazıya dönüştürdükten sonra; metni sunucudaki REST servisine sorgu olarak göndermektedir.
3. **Sunucuda anlamsal analiz aşaması:** Bu aşamada sunucu, girilen cümleyi işleyerek sorguda yapılması istenileni anlar. Burada yapılan işlemleri, işlemin anlaşılması ve parametrelerin çözümlenmesi olarak ikiye ayırabiliriz. Örneğin, "Gökhan’ı ara" sorgusunda, arama servisinin çağrıldığını ve aranılacak kişinin de Gökhan olduğunu anlar. Bunun dışında harici bir servisten bilgi alınmasını gerektiren "İstanbul’da bugün hava nasıl?" sorgusunda; sunucu hava durumu ile



Şekil 2.1: Sistem Çalışma Prensibi

ilgili 3. parti servise, yer olarak İstanbul bilgisini gönderir ve sonucu alır. Bu bölümün detayları bölüm 4'te verilmiştir.

- 4. Sunucunun cevap göndermesi:** İşlenen sorgu sunucu ile mobil uygulama arasında belirlenen protokole uygun bir yapıya dönüştürülür. Bu aşamada düz metin olan sorgu mobil uygulamanın anlayıp işlem yapabileceği bir yapıya dönüştürülür ve sunucu tarafından bu şekilde uygulamaya gönderilir.
- 5. Uygulamanın cevabı anlaması:** Geliştirilen protokol kullanılarak gelen cevap mobil uygulama tarafından anlaşılır ve yapılması gereken işlem ve parametreler çözümlenir.
- 6. İsteğin yerine getirilmesi:** İstenilen işlem için gerekli metotlar parametreleri ile çağrılır ve dönen cevaba göre istek gerçekleştirilir veya gösterilir.

2.1 Sunucu - İstemci Arası Haberleşme

Projemizin istemci-sunucu iletişimi Restful Web Servis mimarisine dayanmaktadır. Restful mimarisinin seçilme sebebi istemci-sunucu arasındaki veri alışverişini SOAP veya RPC gibi karmaşık mimariler yerine, HTTP protokolü üzerinden sağlamak

ve istemci oturumunu tutmayarak (stateless) veri alışverişini daha ölçeklenebilir kılmaktır. Restful servisleri gerçeklemek için sunucu tarafında Spring ve Apache CXF [13] çatıları kullanılmıştır.

Sunucu ile istemcinin aynı dili konuşabilmesi ve aralarındaki iletişimin sağlanması için bir protokol belirlenmiştir. Sunucu tarafında geliştirilen yapı, aşağıda ayrıntıları verilen formatta çıktı üretir ve istemci cihaz da aynı format ile sunucudan gelen yanıtı parçalar ve gerçekleştirir. Geliştirilen iletişim protokolü aşağıdaki bilgileri içermektedir:

- Yanıt Durumu (hatalı, düzgün)
- Yanıt Tipi (servis, yerel)
- İşlem Tipi (arama, mesaj, hava durumu ...)
- İçerik (seçilen servisin gereksinim duyduğu parametreler)

Belirlenen protokolün yapısı aşağıdaki gibidir. Buradaki alanların ne amaçla kullanıldığı çizelge 2.1 üzerinde gösterilmiştir.

```
{
  responseCode: "api/status/ok" | "api/status/error"
  responseType: "native_op" | "service"
  nativeResponse: {

    operationType: "native/call" | "native/sms" |
      "native/email" | "native/search/web" |
      "native/launch" | "native/map" | "native/traffic"

    call: {
      receiverName: ""
      receiverNumber: ""
    }
    sms: {
      bodyText: ""
      receiverName: ""
      receiverNumber: ""
    }
    email: {
      bodyText: ""
      receiverName: ""
      receiverAddress: ""
    }
  }
}
```

```
    search: {
      query: ""
      site: ""
    }
    launch: {
      appName: ""
      param: ""
    }
    map: {
      routeTarget: ""
      startLoc: ""
      isTargetLocality: ""
      targetLocalityLat: ""
      targetLocalityLon: ""
      markerTarget: ""
    }
    traffic: {
    }

    domainType: "weather" | "exchange"

    images: [
      sourceUrl: ""
      title: ""
    ]
    results: [
      text: ""
      subText: ""
      onclickUrl: ""
      imageUrl: ""
    ]
  }
}
```

Örnek olarak "gökhana trafik çok sıkışık mesajını gönder" işlemi için sunucu tarafından dönen yanıt aşağıdaki gibidir.

```
"responseCode": "api/status/ok",
"responseType": "native_op",
"nativeResponse": {
  "operationType": "native/sms",
  "sms": {
    "bodyText": "trafik çok sıkışık",
    "receiverName": "Gökhan"
  }
}
```

Çizelge 2.1: Sunucudan Dönen Cevabın Açıklaması

JSON Alan Adları	Açıklama
responseCode	Başarım kodu
responseType	Servisten gelen yanıtın türü: yerel operasyon yada harici servis
native_op	Cihaz üzerinde başlatılacak yerel işlem türü
service	Harici servis işlemi
call	Arama işlemi
sms	Mesaj Gönderme İşlemi
email	E-posta gönderme işlemi
launch	Mobil cihazda yüklü uygulamayı başlatma
search/web	İnternette arama
traffic	Trafik durumu
map	Harita Servisleri
receiverName	Rehberden işlem yapılacak kişi
receiverNumber	İşlem yapılacak telefon numarası
receiverAddress	İşlem yapılacak e-posta adresi
bodyText	Gönderilecek mesaj ya da e-postanın içeriği
search/site	Arama yapılacak site
search/query	Arama metni
launch/appName	Başlatılacak uygulamanın adı
launch/param	Uygulamaya gönderilecek parametre
map/routeTarget	Variş noktası
map/startLoc	Başlangıç noktası
map/targetLocalityLat	Hedefin enlem bilgisi
map/targetLocalityLon	Hedefin boylam bilgisi
map/isTargetLocality	Hedef belli bir nokta mı yoksa aranacak nesne mi
service/images/source_url	Resim kaynak bağlantısı
service/images/title	Resim başlığı
service/results/text	Sonuç metni
service/results/subtext	Sonuç alt metni
service/results/onclick_url	Sonuç metnine tıklanınca yönlendirilecek kaynak adresi

3. MOBİL UYGULAMA

Proje kapsamında istemci tarafı için Android işletim sistemine sahip telefon ve tabletler üzerinde çalışacak bir uygulama geliştirilmiştir. Bu mobil uygulama sesli sorguları yazıya çevirme ve sunucuya gönderme görevlerinden sorumludur. Ayrıca sunucudan gelen analiz sonucu ile, türüne göre gerçekleştirme veya yazı ve resim gibi görsel öğelerle kullanıcıya sunma işlemlerini gerçekleştirir.

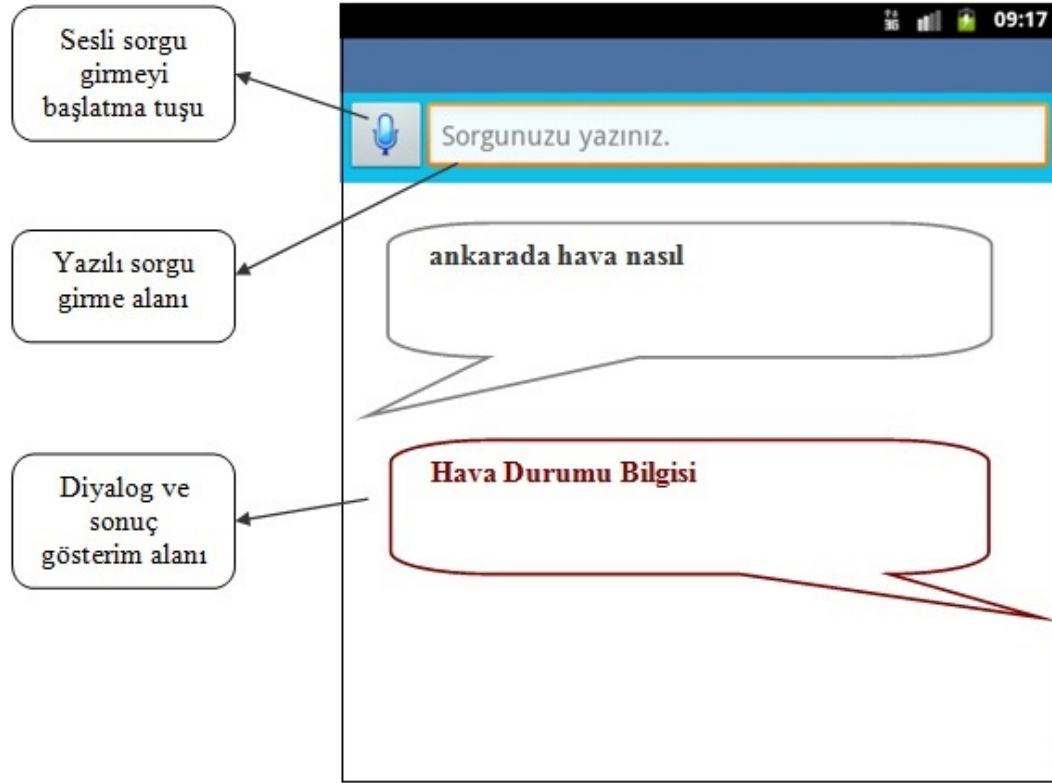
3.1 Uygulama Arayüzü

Şekil 3.1 geliştirilmiş olan mobil uygulamanın (Android) arayüzü göstermektedir. Uygulama genel olarak tek bir ekrandan oluşmaktadır. Üst kısımda sesli sorgu girme tuşu ve yazılı sorgu girme alanı bulunmaktadır. Bu kısmın altındaki kısım ise sonuç gösterim alanıdır. Şekil 3.2 kullanıcıdan sesli komut dinleme ekranı çıktılarını göstermektedir.

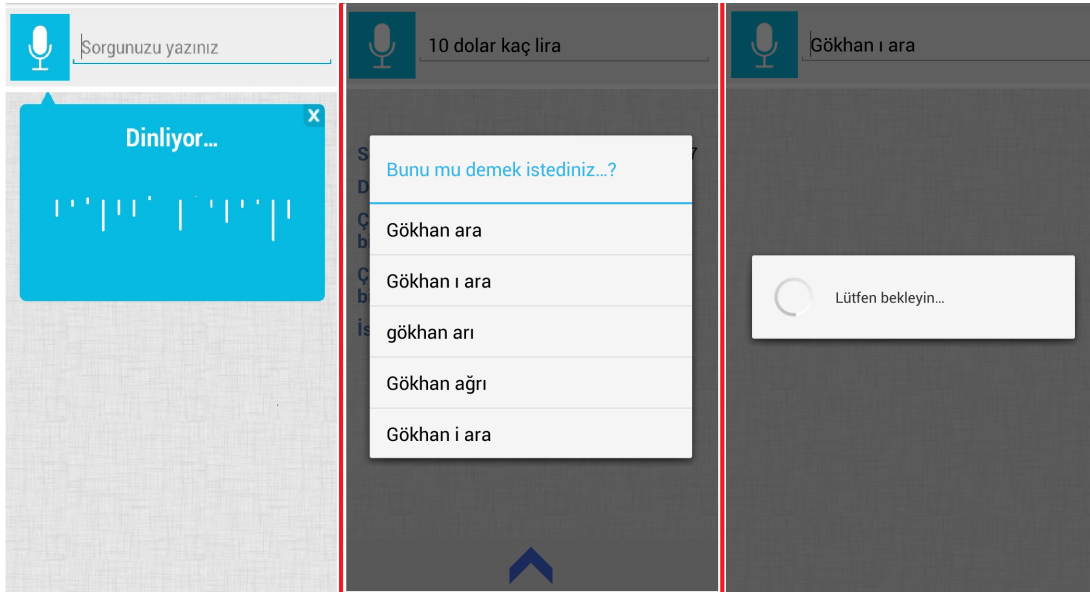
Sonuç gösterme alanı anlamsal çözümleme servisinin yanıtına göre dinamik olarak oluşturulur; metinsel yanıtlar gerektiğinde resim ve harita ile birlikte verilir. Dönen yanıt sadece düz metinden oluşuyorsa tek bir alanda, birden fazla maddeden oluşan bir metinse alt alta verilir. Gerektiğinde bu yanıt alanları sonuçların alındığı kaynak adreslerini ya da bulunabileceği arama bağlantı dizinlerini de taşır. Yanıt resim ya da harita içerdiğinde bu alanlar metinsel yanıtların üstünde verilir. Dönen yanıt mobil cihaz üzerinde başka bir uygulamanın başlatılmasını gerektiriyorsa, uygulama, gerekli parametreleri vererek bu uygulamayı başlatır.

3.2 Uygulamanın Çalışma Akışı

Uygulamanın genel çalışma prensibi Şekil 3.3’de gösterilen akış diyagramındaki gibidir. Sesli sorgu girmek için kullanıcı mikrofon imgesi bulunan tuşa basar, açılan dinleme ekranı vasıtası ile de sorgusunu girer. Hali hazırda bu mobil uygulama Google’ın konuşmadan metine dönüşüm servisini [12] kullanmaktadır. Bu servis

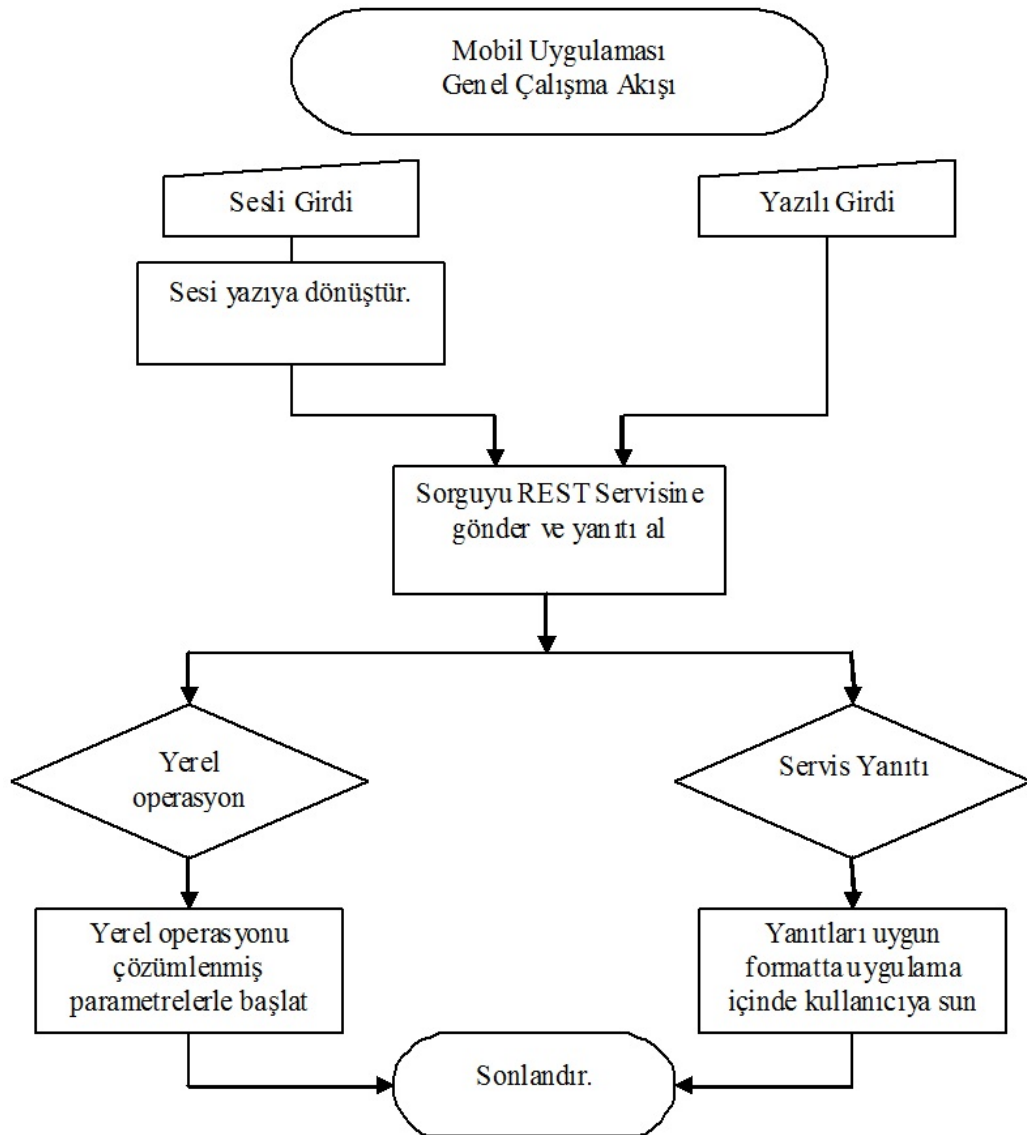


Şekil 3.1: Mobil Uygulama Arayüzü

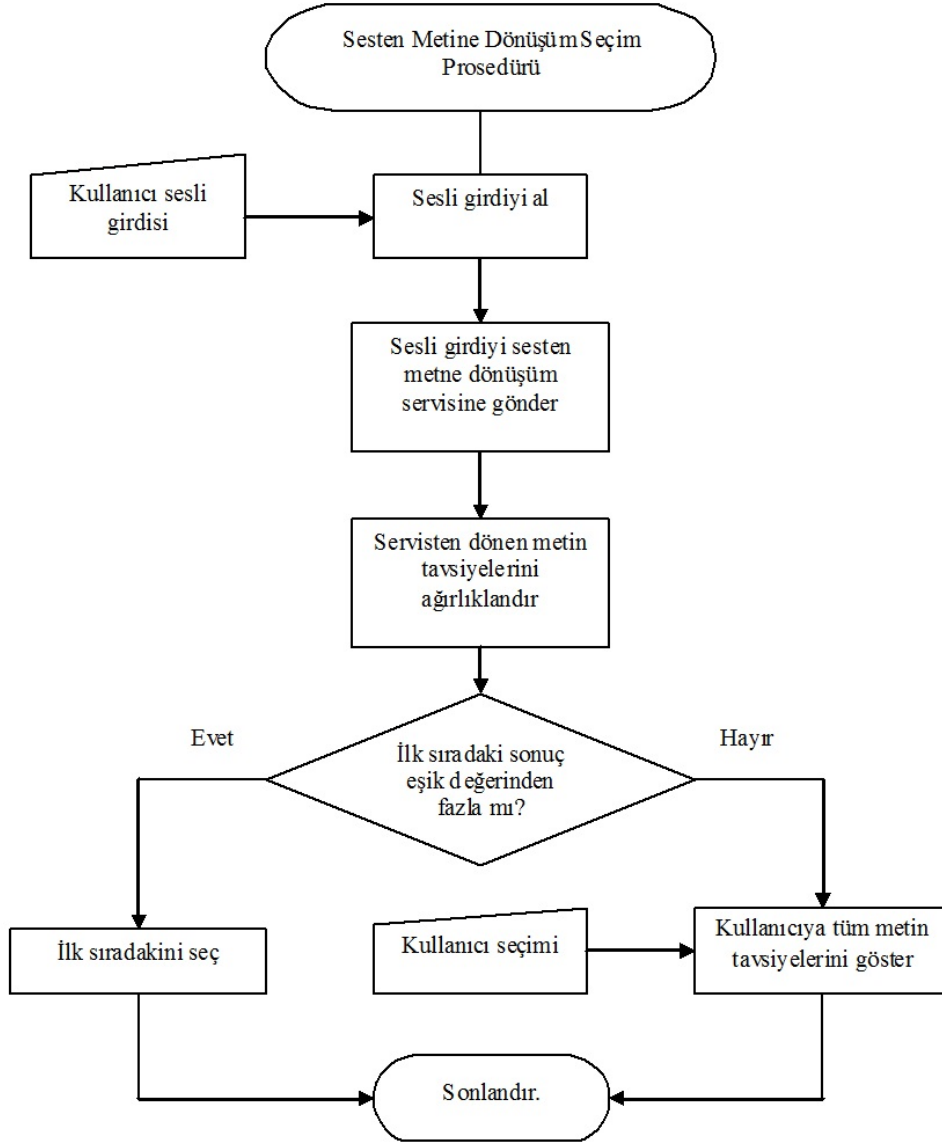


Şekil 3.2: Mobil Uygulama Ses Dinleme

Türkçe dahil, pek çok dil için sesli girdileri eşleştirdiği en yakın söz öbeklerine çevirir. Uygulamamız bu önerileri kullanıcıya belirsizliği ortadan kaldırma amacıyla sunar, kullanıcı da bu önerilerden bir tanesini seçerek sorgusunun REST servislerine gönderilmesini sağlar. Google servisinden birden fazla öneri, güvenilirlik oranı ile birlikte dönmektedir. Dönen önerilerden belirlenen eşik değerin (0.7) üzerinde güvenilirlikte olan varsa, otomatik olarak seçilir, yoksa kullanıcıya istediği çeviriyi seçmesi için bir liste sunulur. Planlanan bu sürecin ekran görüntüleri Şekil 3.2’de akışı Şekil 3.4’te gösterilmiştir.



Şekil 3.3: Mobil Uygulama Çalışma Akışı



Şekil 3.4: Sesten Metne Dönüşüm Seçim Prosedürü

Sesle sorgu girebilmenin yanı sıra uygulama aynı zamanda yazı ile sorgu girme imkanı da tanımaktadır. Sonuç gösterim alanı, kullanıcının sorgularını ve servisten dönen yanıtları uygun şekilde gösterir.

Kullanıcı yazılı ya da sesli sorgusu alındıktan sonra, servisle haberleşme üç adımda gerçekleşir. Öncelikle istemci kullanıcı bağlamını oluşturur; bu bağlam içinde kullanıcının kimliği, konum ve zaman bilgileri bulunur. İkinci aşamada bağlam ile birlikte sorgu sunucu servise HTTP POST isteği olarak gönderilir. Son aşamada ise sunucudan gönderilen sorgunun yanıtı alınır. Örnek bir istek şu şekildedir:

URL Adresi: IP_Adres:Port/SanalBilge/rest/info

İçerik:

```
{
  "username": "android",
  "query": "gökhani ara",
  "latitude": 41.00527,
  "longitude": 28.97696,
  "timestamp": 1427735981599
}
```

İstemciden REST servisine sorgu ile birlikte gönderilen parametreler Çizelge 3.1’de verilmiştir. Bunlardan "Username", "Query" ve "Timestamp" her sorguda bulunurken, konum parametreleri ise mümkün olduğunda¹ eklenmektedir. Konum bilgisi ve zaman parametreleri, servisin kullanıcı bağlamını hesaba katarak daha başarılı yanıt üretmesini sağlamak amacıyla gönderilir.

Çizelge 3.1: Gönderilen İsteğin Parametreleri

HTTP POST Parametreleri	Açıklama
Username	Kullanıcıyı tanımlayan değer
Query	Sorgu metni
Latitude	Kullanıcının bulunduğu yerin enlem değeri
Longitude	Kullanıcının bulunduğu yerin boylam değeri
Timestamp	İstek gönderilme zamanı (UTC formatında)

Sorulan sorgular ya da istekler REST servisleri tarafından iki temel kategoriye ayrıştırılır: yerel operasyonlar ve servis yanıtları. Yerel operasyonlar mobil cihaz üzerinden gerçekleştirilecek istekleri temsil eder. Yerel operasyon isteklerinde

¹GPS alıcısının açık olduğu durumlarda

sunucunun görevi, işlemi anlamak ve işlemin olası parametrelerini doğru ayrıştırıp istemciye geri bildirmektir. Servis yanıtları ise sorguların ya da isteklerin yerine getirilebilmesi için sunucunun dahili ve harici servislerin kullanılması gerektiği durumları temsil eder. Bu durumda sunucunun görevi doğru servisleri kullanarak kullanıcıyı tatmin edici yanıtları veya sonuçları üretebilmektir.

3.3 Desteklenen Operasyonlar

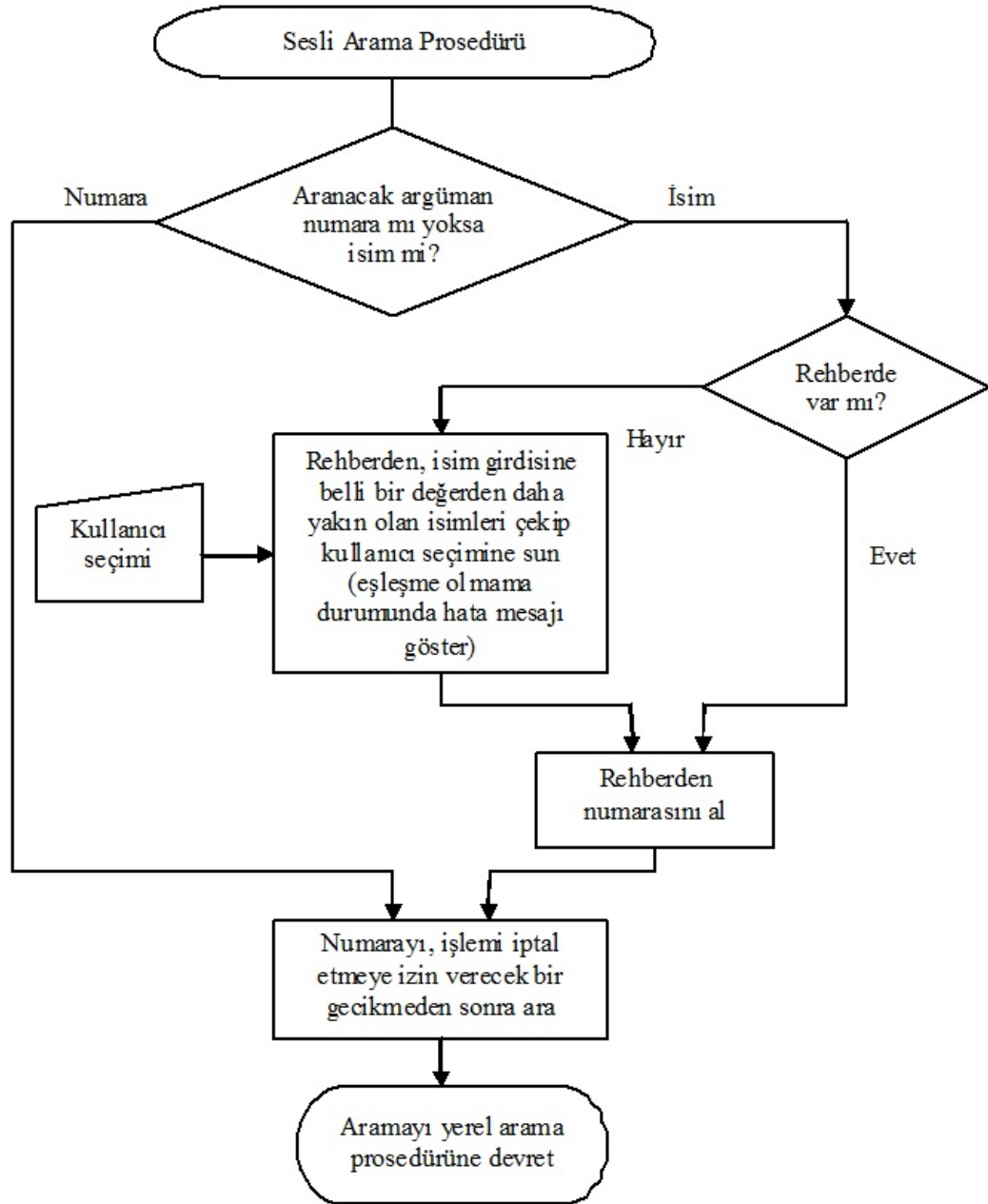
Geliştirilen mobil uygulama hali hazırda, sesli arama, mesaj ve e-posta gönderme, uygulama başlatma, trafik gösterme, haritadan yol tarifi, internette arama, döviz ve hava durumu görüntüleme operasyonlarını desteklemektedir.

Rehberde kişiler ad ve soyad olarak kaydedildiğinden isimler hem isim-soy isim beraber hem de ayrı ayrı indekslenir.

Telefon rehberindeki kişilerin birden fazla adı veya soyadı olabilir, bunun dışında cihazda yüklü olan uygulamalar da birden fazla kelimedden oluşabilir ve kullanıcı sorgusu bu yapıya tam olarak uymayabilir. Örneğin telefon rehberinde "Ali İsmail" şeklinde kayıtlı bir kişiyi aramak için kullanıcı yalnızca "İsmail'i ara" diyebilir. Bu sorunu çözmek için aranan kelime ile tam eşleşme bulunamaması durumunda Monge-Elkan yaklaşık metin eşleme yöntemi [14] ile birden fazla kelime ihtimalini göze alarak arama yapılır. Bu yöntemin Levenshtein [15] mesafesi yöntemine göre temel avantajı birden fazla kelime içinde arama yapacağı zaman da çok farklı olduklarını iddia etmemesidir. Bu algoritma bize olası eşleşmeleri benzerlik puanları ile birlikte içeren bir liste döndürür. Bu listede tanımlanan eşik değerin üzerinde bir eşleşme varsa, doğru olarak seçilir ve işlem devam eder. Aksi durumda kullanıcıya benzerlik puanına göre sıralanmış bir liste gösterilir ve doğru eşleşmeyi elle seçmesi beklenir.

3.3.1 Sesli arama

Sesli arama servisi, kullanıcının, rehberinde bulunan bir kişiyi ya da bir numarayı araması işlemini gerçekleştirir. Mobil cihaz üzerinde arama akışı Şekil 3.5'deki akış diyagramda gösterilmiştir.



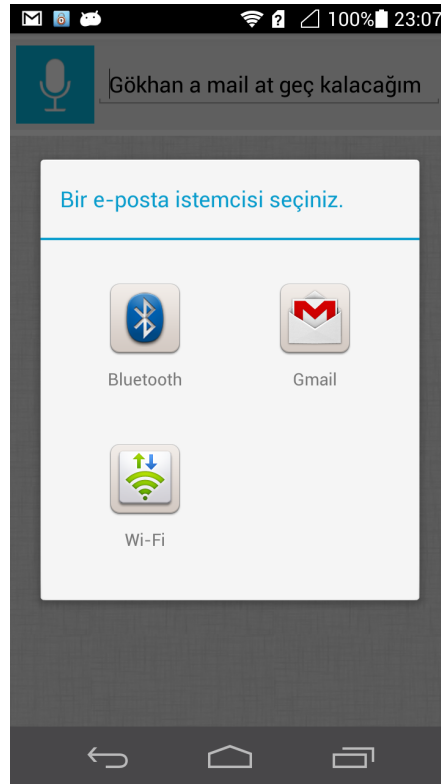
Şekil 3.5: Sesli Arama Akışı

3.3.2 Mesaj gönderme

Mesaj gönderme servisi sesli arama fonksiyonuna benzer bir akış yapısına sahiptir. Farkı, arama yerine mesaj gönderme ile anlamsal olarak ilişkili komutların söylenmesi ya da yazılmasıyla başlatılmasıdır. Ayrıca, kullanıcı mesaj olarak gönderilecek metin bilgisini iletmişse bu bilgi cihazın yerel mesaj gönderme servisine parametre olarak verilir.

3.3.3 E-Posta gönderme

E-Posta gönderme servisi, kullanıcının isteğinin anlaşılıp uygulamaya bildiriminden sonra cihazda etkin olan e-posta istemci uygulamasının başlatılması ve varsa çözümlenen içeriğin yerel servise parametre olarak verilmesiyle çalışır. Şekil 3.6 mobil cihazda yüklü e-posta uygulamaları listeleyen ve kullanıcıdan istediğini seçebilmesini sağlayan ekran görüntüsünü göstermektedir.



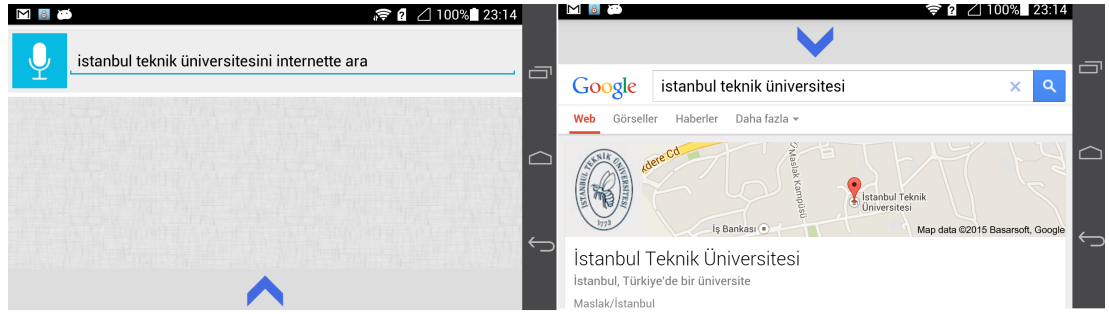
Şekil 3.6: Uygun E-Posta Uygulaması Seçme Ekranı

3.3.4 Uygulama başlatma

Uygulama başlatma servisi, kullanıcının, mobil cihazında kurulmuş olan bir uygulamayı başlatır. Sesli arama ve mesaj gönderme servislerinde rehberde verilen ismi bulma ya da benzetme yapısı cihazda yüklü olan uygulama isimleri için kullanılmaktadır.

3.3.5 İnternette arama

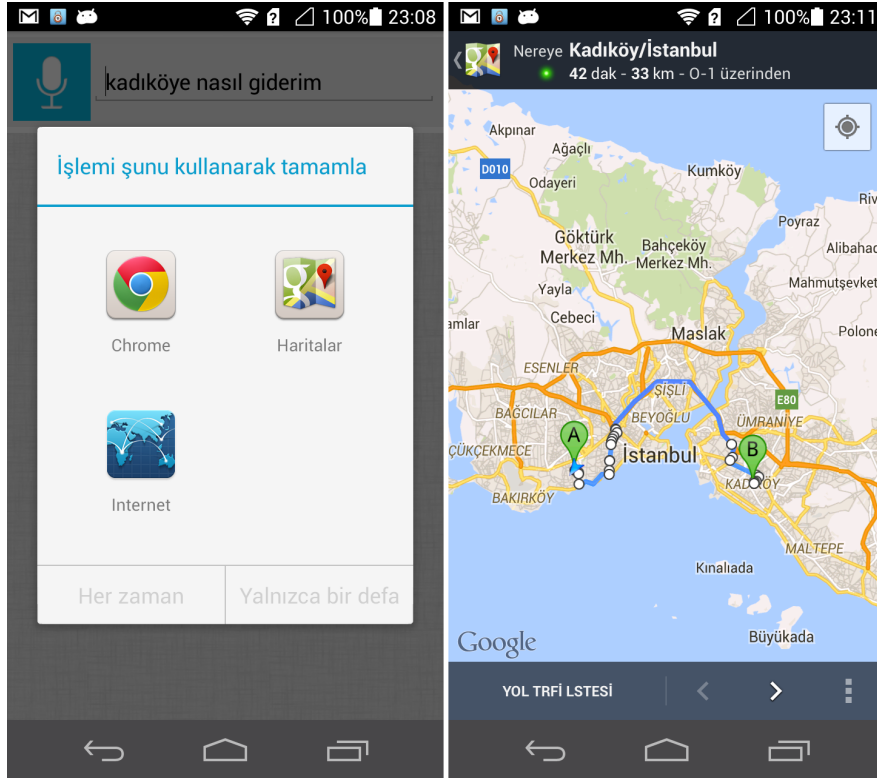
İnternette arama servisi, sunucudan dönen yanıtta yer alan arama metnini internette arar. Sonuçlar kullanıcıya gömülü bir internet tarayıcısında gösterilir. Şekil 3.6 bu adımın ekran çıktısını göstermektedir.



Şekil 3.7: İnternette Arama İşlemi Ekran Görüntüsü

3.3.6 Haritada arama

Mobil cihazda yüklü olan harita uygulamalarından hangisini kullanmak istediği kullanıcıya sorulur. Başlangıç-bitiş noktaları, hedefin enlem boylam değerleri, hedefin belirli bir bitiş noktası mı yoksa aranılacak bir nokta mı olduğu bilgilerinden var olanlar seçilen uygulamaya parametre olarak gönderilir. Sunucudan dönen cevapta başlangıç değeri bulunmuyorsa mevcut konum başlangıç noktası olarak seçilir. Şekil 3.8 bu adımın ekran çıktısını göstermektedir.



Şekil 3.8: Harita İşlemi Ekran Görüntüsü

3.3.7 Trafik bilgisi

Yandex Trafik servisi² kullanılarak bulunulan konumun etrafındaki trafik durumu gösterilir. Şekil 3.9(a) bu adımın ekran çıktısını göstermektedir.

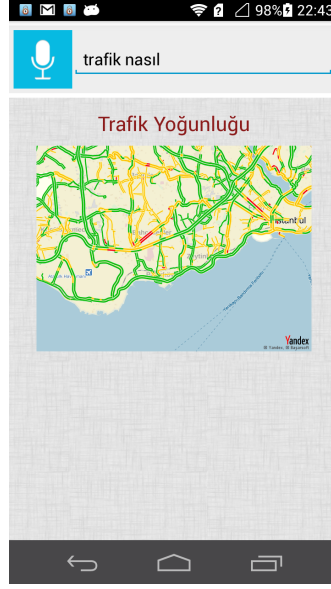
3.3.8 Hava durumu görüntüleme

Sunucudan gelen yanıt doğrultusunda hava durumu bilgisi, konum, varsa resimleri gösterilir. Şekil 3.9(b) bu adımın ekran çıktısını göstermektedir.

3.3.9 Döviz bilgisi

Sunucudan gelen yanıt doğrultusunda, çevrilmek istenilen ve çevrilecek para birimi, miktar ve toplam tutar listelenir. Şekil 3.9(c) bu adımın ekran çıktısını göstermektedir.

²<http://trafik.yandex.com.tr>



(a) Trafik Durumu



(b) Hava Durumu Bilgisi



(c) Döviz Bilgisi

Şekil 3.9: Mobil Uygulama Ekran Görüntüleri

4. SORGUNUN ANLAŞILMASI ve SUNUCU YAPISI

Mobil cihaz tarafından yazılı hale dönüştürülen kullanıcı isteği REST API teknolojisi kullanılarak sunucuya iletilir. Tezin bu bölümünde düz metin halinde gelen sorguların bilgisayarın anlayabileceği bir sorguya nasıl dönüştürüldüğü anlatılmıştır. Şekil 4.1’de sunucu tarafında gerçekleştirilen işlem akışı gösterilmiştir.

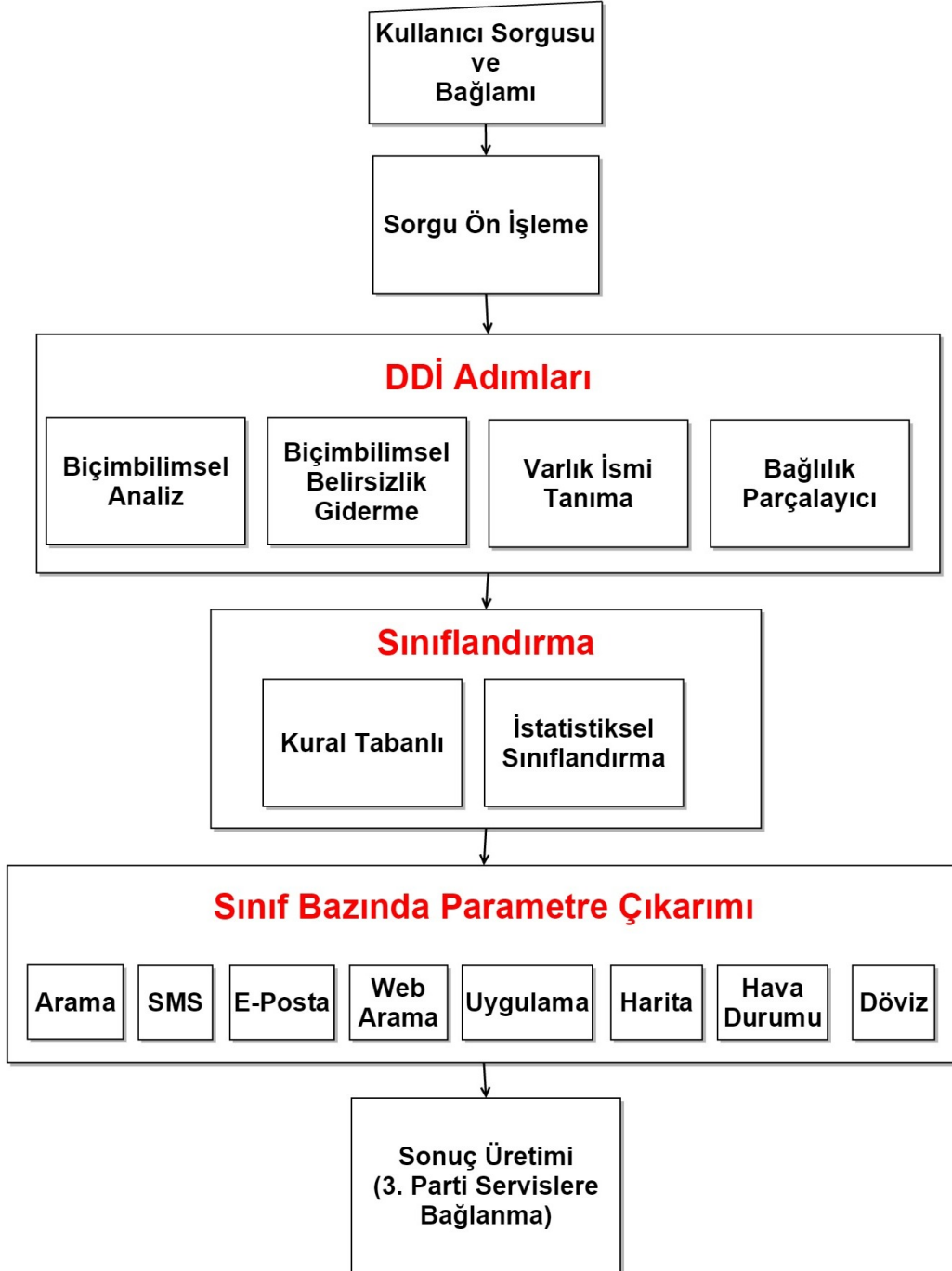
Yapılan işlemleri sıralamak gerekirse; ilk etapta, gelen sorgudaki olası dil bilgisi hatalarını gidermek için normalizasyon yapılır. Ardından sonraki adımlarda kullanılmak üzere Doğal Dil İşleme (DDİ) araçları uygulanır. DDİ adımıyla biçimbilimsel analiz, Varlık İsmi Tanıma (NER) ve bağıllık ayrıştırıcı kullanılır.

Bu aşamalardan sonra sorgunun hangi servis ile ilgili olduğu belirlenir. Bu doğrultuda öncelikle daha önceden belirlenmiş kurallardan herhangi biri ile uyuşup uyuşmadığı kontrol edilir. Eğer sorgu bir kural ile eşleşirse o kuralın ait olduğu servis seçilir, sorgu herhangi bir kurala uymazsa istatistiksel bir yöntem kullanılarak ilgili çözümleme servisi seçilir.

Sorgu sınıflandırma işleminden sonra seçilen servisin çözümleyicisine yönlendirilir. Çözümleme aşamasında servis için gerekli olan parametreler çıkartılır. Örneğin, "Gökhan'ı ara" sorgusu, arama servisine yönlendirildikten sonra; aranacak kişinin "Gökhan" olduğunun anlaşılması gerekmektedir.

4.1 Sorgu Ön İşleme

Çalışılan alan konuşma verisi olduğundan, dil bilgisi açısından hatalı sorgular geldiği gözlenmiştir. Ayrıca kullanılan ses tanıma aracı da hatalı çıktılar vermektedir. Bu hataların en önemlisi özel isimlerde tüm karakterlerin küçük harfli olması ("gökhan") ve çekim eklerinin ayrı bir kelime olarak gelmesidir ("gökhan a"). Bu adımda, sorgudaki bahsedilen imla ve yazım hataları düzeltilmeye çalışılmıştır. Belirtilen bu



Şekil 4.1: Sunucu Tarafındaki Sistem Akışı

hatalar DDİ araçlarının performansına ciddi zarar vermekte ve sonraki adımlardaki başarıyı düşürmektedir.

Dil bilgisi hatalarının çözümü için İstanbul Teknik Üniversitesi Doğal Dil İşleme Grubu tarafından geliştirilen normalizasyon (<http://tools.nlp.itu.edu.tr/Normalization>) aracı [16] incelenmiştir. Bu araç büyük-küçük harf düzeltme, kelime değiştirme, sözlük listesi, özel isim tespiti, Türkçe harf kurtarma, ünlü harf kurtarma, aksan düzeltme işlemlerini gerçekleştirebilmektedir. Geliştirilen uygulama bu özelliklerden yalnızca büyük-küçük harf ve aksan düzeltme işlemlerine ihtiyaç duymuştur. Bu yüzden sadece belirtilen 2 özelliği içeren bir normalizasyon modülü geliştirilmiştir. Ayrıca ayrı kelime olarak işaretlenen çekim ekleri hatalarının çözümü de bu modüle eklenmiştir. Türkçe çekim eki listesi tanımlanmış, bu listede bulunan ekler, kendisinden önceki kelime ile birleştirilmiştir. Eğer bir önceki kelime özel isim ise bu birleştirme işlemi araya kesme işareti (') konularak yapılmıştır.

4.2 DDİ Araçları

Doğal Dil İşleme adımıyla biçimbilimsel analiz [17] ve varlık ismi tanıma (NER) [6,18] araçları doğrudan kullanılmış, bağımlık ayrıştırıcı (Dependency Parser) [19] aracı alan adaptasyonu uygulanarak kullanılmıştır.

4.2.1 Varlık ismi tanıma

Tez kapsamında konuşma metinlerine alan adaptasyonu yapılmış bir varlık ismi tanıma aracı geliştirilmiştir. Ancak bu teze paralel olarak yapılan bir çalışmadan ortaya çıkan NER aracının [18] daha yüksek başarımlar verdiği gözlemlendiğinden nihai sistemde bu araç ile yola devam edilmiştir.

Varlık ismi tanıma adımıyla, cümledeki varlıklar belirlenir. Geliştirilen algorithma Kişi, Yer, Kurum, Zaman, Yüzdelik İfade ve Para varlıkları desteklenmektedir. Örnek olarak NER aracı "Yarın Gökhan Çelikkaya ile Ankara'ya gideceğim." cümlesini "Yarın=ZAMAN", "Gökhan Çelikkaya=KİŞİ" ve "Ankara=YER" olarak işaretler. Bu adımın çıktıları sorgu sınıflandırma ve parçalama adımlarında kullanılmaktadır.

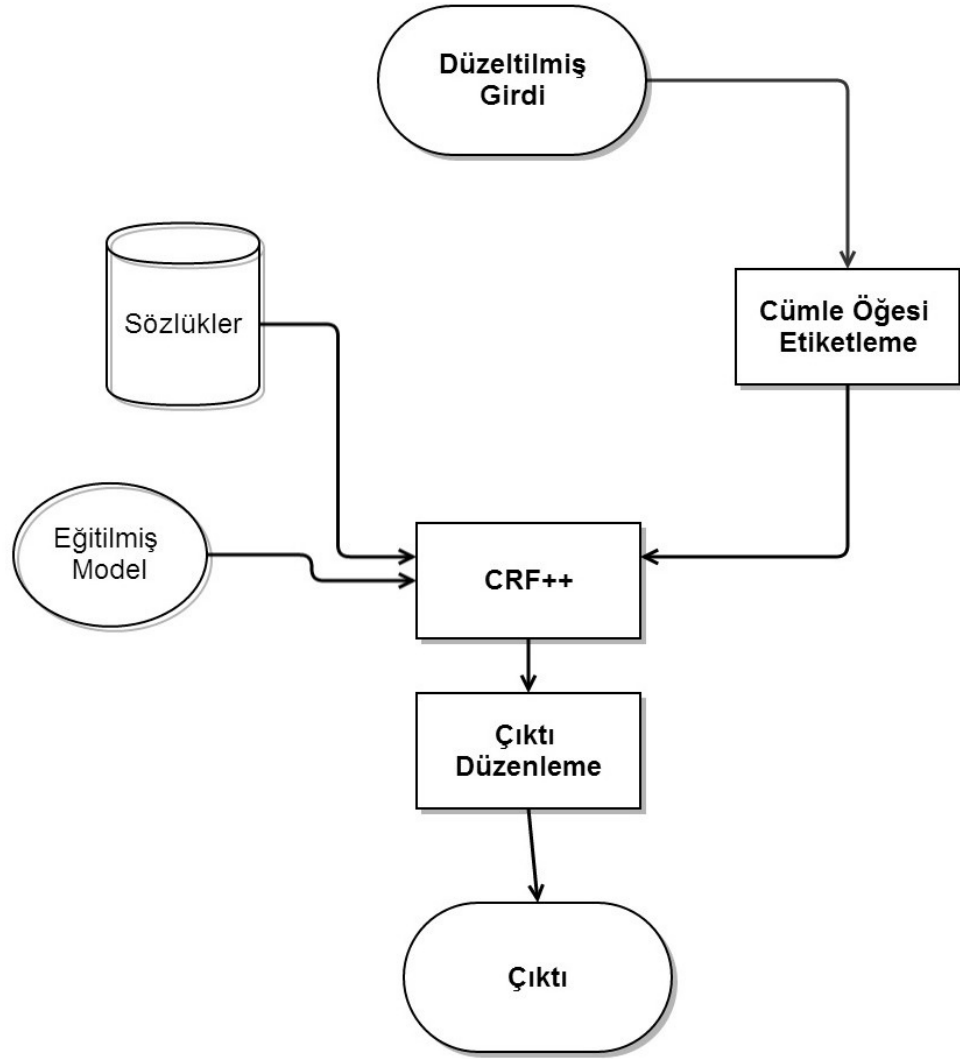
Geliştirilen NER algoritması istatistiksel bir yöntemle dayanmaktadır ve gazetelerden toplanan 750 cümlelik veri ile eğitilmiştir. Eğitim adımında makine öğrenmesi algoritması olarak bir dizilim sınıflandırıcısı olan koşulla rastgele alanlar (CRF) [20] kullanılmıştır. Bu algoritmanın seçilme nedeni, tek bir kelime yerine önceki ve sonraki kelimeleri, yani cümle dizilimini de kullanarak sınıflandırma yapmasıdır. CRF algoritması, Saklı Markov Modeller (HMM) [21], stokastik gramer ve Maksimum Entropi Markov Modellerin (MEMM) avantajlarına sahiptir.

Tez kapsamında aracın geliştirilmesi adımında CRF++ [22] aracı kullanılmıştır. Aracın eğitimi için özellikler arasındaki ilişkileri belirten bir şablon dosyası oluşturulmuştur. Proje çalışması esnasında gelen sorgu cümlesi eğitim için hazırlanan veri ile aynı formata çevrilir ve üretilmiş olan model dosyası kullanarak her kelime için sınıflandırma yapılır. Şekil 4.2 geliştirilen NER aracının akışını göstermektedir.

Şekil 4.3’de NER aracımızı eğitmek için kullandığımız veriden örnek bir parça gösterilmiştir. Oluşturulan modelde kullanılan özellikler aşağıdaki gibidir.

1. Kelimenin tamamı
2. Kelimenin gövdesi
3. Kelime türü
4. İsmi hali
5. Özel isim
6. Kelimenin çekim ekleri
7. Büyük-küçük harf durumu
8. Cümle başında olma durumu
9. Sözlüklerde olma durumu

Oluşturulan sözlükler **temel** ve **üretken** olmak üzere iki yapıdan oluşmaktadır. Temel sözlükler kişi, yer ismi doğrudan varlığı belirtirken üretken sözlükler varlıklardan önce



Şekil 4.2: Varlık İsmi Tanıma Aracı ile İşaretleme Akışı

sabri	Sabri	Noun	NOM	1	Prop+A3sg+Pnon+Nom	2	0	1	1	0	0	0	0	0	0	PERSON
ile	ile	Postp	NONE	0	PCNom	0	0	0	0	0	0	0	0	0	0	
volkan	volkan	Noun	NOM	0	A3sg+Pnon+Nom	2	0	1	1	0	0	0	0	0	0	PERSON
arasındaki	ara	Adj	NONE	0	Rel	0	0	0	0	0	0	0	0	0	0	
tartışmanın	tartış	Noun	GEN	0	Inf2+A3sg+Pnon+Gen	0	0	0	0	0	0	0	0	0	0	O
da	da	Conj	NONE	0	0	0	0	1	0	0	0	0	0	0	0	
devre	devre	Noun	NOM	0	A3sg+Pnon+Nom	0	0	0	0	0	0	0	0	0	0	O
arasında	ara	Noun	LOC	0	A3sg+P3sg+Loc	0	0	0	0	0	0	0	0	0	0	
başladı	başla	Noun	NOM	0	PastPart+A3sg+P3sg+Nom	0	0	0	0	0	0	0	0	0	0	O
vurgulandı	vurgula	Verb	NONE	0	Pass+Pos+Past+A3sg	0	0	0	0	0	0	0	0	0	0	O
.	.	Punc	NONE	0	0	0	0	0	0	0	0	0	0	0	0	
ulusal	ulusal	Adj	NONE	0	0	2	0	0	1	0	0	0	0	0	0	ORGANIZATION
felaket	felaket	Noun	NOM	0	A3sg+Pnon+Nom	2	0	0	0	0	0	0	0	0	0	ORGANIZATION
önleme	önle	Noun	NOM	0	Inf2+A3sg+Pnon+Nom	2	0	0	0	0	0	0	0	0	0	ORGANIZATION
merkezi	Merkez	Noun	ABL	1	Prop+A3sg+P3sg+Abl	2	0	0	0	1	0	0	0	0	0	ORGANIZATION
'nden	'nden	kesme	NONE	0	kesme	2	0	0	0	0	0	0	0	0	0	
yapılan	yap	Adj	NONE	0	PresPart	0	0	0	0	0	0	0	0	0	0	
açıklamada	açıklamada	Noun	NOM	1	Prop+A3sg+Pnon+Nom	0	0	0	0	0	0	0	0	0	0	O

Şekil 4.3: Varlık İsmi Tanıma Aracı Eğitiminde Kullanılan Veri Örneği

veya sonra gelen unvan gibi kelimeleri kapsamaktadır. Bu yapılar aşağıda belirtilen sözlükleri içermektedir.

Temel sözlükler: Ad, Soyad, Yer, Zaman, Para Birimi

Üretken sözlükler: Unvan, Yer Türeteç, Kurum Türeteç

4.2.2 Bağlılık ayrıştırıcı

MaltParser [19,23] veri odaklı çalışan bir bağlılık ayrıştırıcısıdır. Bu yapı ayrıştırıcısının birçok dilde kullanılmasını yani dilden bağımsız olmasını sağlamıştır. Bağlılık analizi yapılmış ve ön işlemlerden geçirilip işaretlenmiş olan bir veri bankası sayesinde eğitilerek kendi ayrıştırıcı modelini oluşturan bu yöntem, daha sonra bağlılık analizi yapılmak istenilen cümleler için kullanılır. Geleneksel yaklaşımlar önceden verilen bir dil bilgisi üzerinden çalışırken, MaltParser veri odaklı olduğu için öğrenme aşamasında, işaretlenmiş olan bir veri bankası kullanır. Proje kapsamında 620 cümlelik bir veri işaretlenmiş ve bu veri aracın eğitiminde kullanılmıştır.

MaltParser öğrenme ve ayrıştırma olmak üzere iki şekilde çalışmaktadır. Öğrenme yönteminde koşturulan araç veri bankasındaki işaretlenmiş veriden yararlanarak kendine bir öğrenme modeli oluşturur. Makine öğrenmesi kullanılarak üretilen bu model, ayrıştırma yönteminde kelimeler arası ilişkilerin sınıflandırılması için kullanılır. Gönderilen girdi içerisindeki her bir cümle için içerisinde bulunan kelimelerin ilişkilerini belirten bir bağlılık grafiği oluşturulur. Bu grafik üzerindeki sonuçlar içerisinden uygulamamız için gerekli olan alanlar kendimizin hazırladığı bir başka ayrıştırıcı ile elde edilir.

Uygulamamız için gelen cümle, biçimbilimsel analizden sonra MaltParser'la uyumlu olan bir özellik modeline uygun olacak şekilde biçimlendirilir. Bu aşamada DDİ çalışmalarında yaygın olarak kullanılan CoNLL [24] veri formatına dönüşüm sağlanır. Bu veri formatı aşağıda görüldüğü gibi belirli alanları içerir. Alan isimleri, kategorileri ve tiplerini içeren bu modelde alan tanımları kısaca aşağıda yapılmıştır ve Çizelge 4.1 "Antalya'da hava nasıl" cümlesi için örnek bağlılık analizini göstermektedir.

```
<dataformat name="conllx">
  <column name="ID" category="INPUT" type="INTEGER"/>
  <column name="FORM" category="INPUT" type="STRING"/>
```

```

<column name="LEMMA" category="INPUT" type="STRING"/>
<column name="CPOSTAG" category="INPUT" type="STRING"/>
<column name="POSTAG" category="INPUT" type="STRING"/>
<column name="FEATS" category="INPUT" type="STRING"/>
<column name="HEAD" category="HEAD" type="INTEGER"/>
<column name="DEPREL" category="DEPENDENCY_EDGE_LABEL"
type="STRING"/>
<column name="PHEAD" category="IGNORE" type="INTEGER"
default="_"/>
<column name="PDEPREL" category="IGNORE" type="STRING"
default="_"/>
</dataformat>

```

Çizelge 4.1: "Antalya'da hava nasıl" cümlesi için bağılılık analizi örneği

ID	1	2	3	4
Form	Antalya'da	hava	nasıl	bugün
Lemma	Antalya	hava	nasıl	bugün
Cpostag	Noun	Noun	Adverb	Noun
Postag	Prop	Noun	Adverb	Noun
Feats	A3sglPnonlLoc	A3sglPnonlNom	_	A3sglPnonlNom
Head	3	3	0	3
Deprel	MODIFIER	OBJECT	PREDICATE	MODIFIER
Phead	_	_	_	_
Pdeprel	_	_	_	_

ID, her bir kelime için verilen farklı bir tam sayı değerini içerir. Form kelimenin kendisini ya da noktalama işaretini gösterir. Lemma ise kelimenin kök halini içerir. CPostag alanı da işaretlenmiş olan kelimenin türünü gösterir. Postag alanı ise türü detaylandırılmış olarak gösterir ve eğer detay içermiyorsa CPostag alanı ile aynıdır. Feats alanında kelimenin olası olan tüm söz dizimsel ve biçimbilimsel özellikleri aralarında (mantıkta "veya" operatörü olan) "|" kullanılarak sıralanır. Head alanında ise o cümle için kelimenin ilişkilendirildiği kelimenin ID'si gösterilir. Deprel alanında ise iki kelime arasındaki ilişkinin türü belirtilir. PHead ve PDeprel alanları ise kategori tipinden de görüldüğü gibi önemsiz alanlardır. "_" işareti ile gösterilir.

4.3 Sorgu Sınıflandırma

Kullanıcının girdiği sorguların hangi servislere yönlendirileceğine karar vermek için iki aşama bulunmaktadır. DDİ araçlarından geçmiş sorgu, ilk önce önceden tanımlanmış kurallar ile karşılaştırılır. Sorgu bu kurallardan birine uymuyorsa, makine

öğrenmesi algoritmaları ile daha önceden eğitilmiş bir sınıflandırıcı ile doğru servis bulunmaya çalışılır. Şekil 4.4'te sınıflandırma adımının algoritması gösterilmiştir.

4.3.1 Kural tabanlı yöntemler

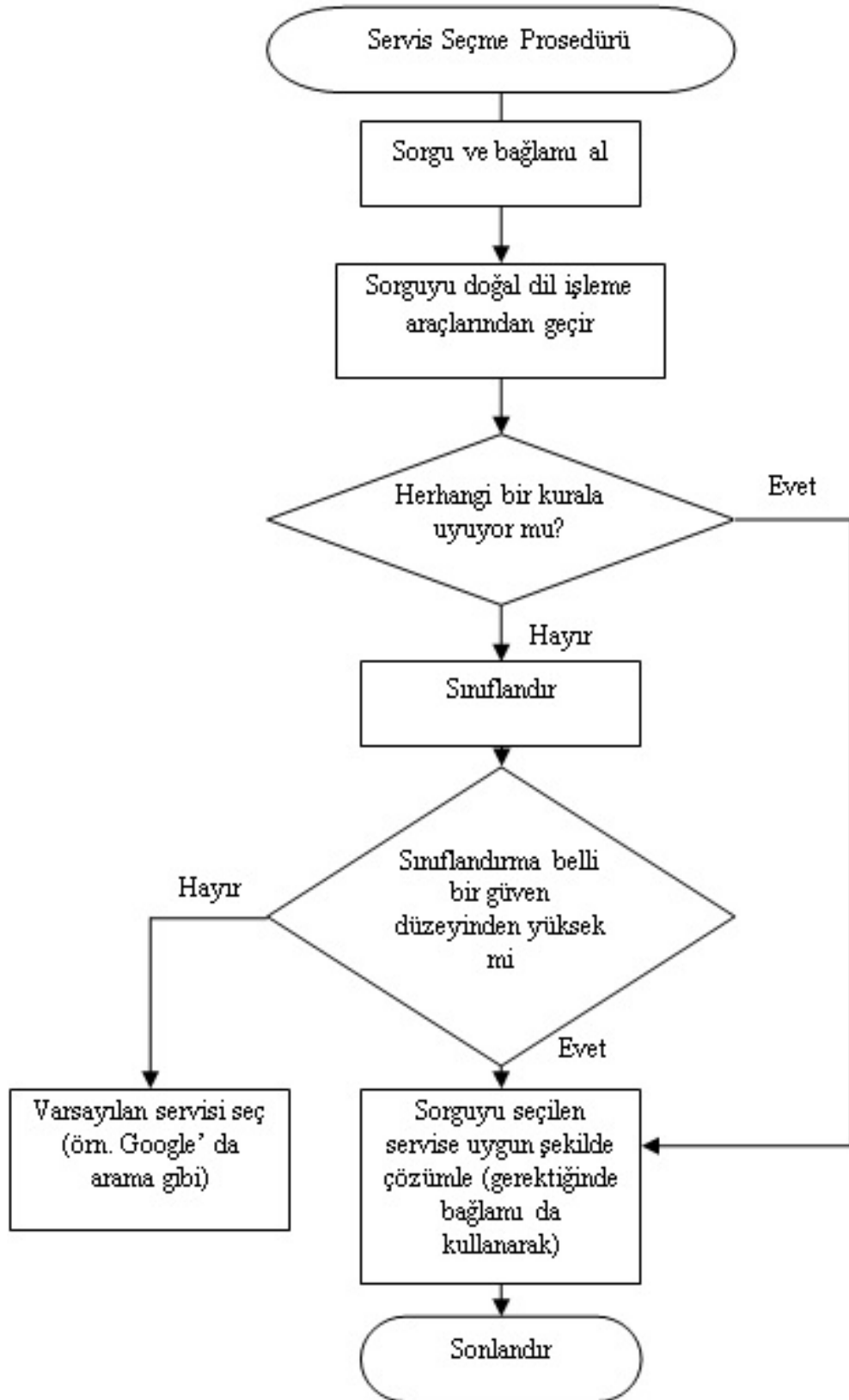
DDİ araçları ile işlenmiş olan sorgular, önceden belirlenmiş olan kurallara göre kontrol edilirler ve uyması durumunda ilgili servise yönlendirilir. Her bir kural, o kurala uyması için gereken dizilimi, kuralın nasıl kontrol edileceğini ve uygun sorgunun hangi servise yönlendirileceğini ve kuralın öncelik değerini içerir.

Bu sistem kapsamında geliştirdiğimiz kural motoru, çok sayıda ve kolayca kural eklenmesini sağlamaktadır. Kurallar en temelde kelime dizilimlerine göre kontrol edilir ve düzenli ifadeler” (regex) [25] ile çok esnek bir biçimde belirlenebilir. Örneğin, "mesaj gönder" cümlesinde kelime dizilimi cümlenin başında geçiyorsa, bunu mesaj gönderme servisine yönlendirebiliriz. Bu kural düzenli ifadeler kullanılarak genişletilebilir. "Lütfen mesaj gönder" cümlesi de aynı anlamı taşıdığı için, koşullu olarak kuralın başına "lütfen" ifadesi eklenebilir. Aynı şekilde "Gökhan'a mesaj gönder" sorgusu da aynı servise yönlenecektir. Her ismi tek tek belirtmeyeceğimiz için NER etiketlerinden yararlanarak kuralı "<Person> eposta gönder" şekline dönüştürebiliriz. Buradaki "<Person>" da isteğe bağlı yazılabilir. Bu örnek kuralın yazımı aşağıdaki gibidir.

Sorgu, Kontrol Tipi, Servis, Öncelik

```
^(Lütfen)? (<Person>)? (eposta|email) gönder, NER_TEXT, SMS, 1
```

"^" işareti bu dizilimin cümle başından başlaması gerektiğini gösterir. "?" işaretinin anlamı ise parantez içine aldığı dizilimin opsiyonel olduğudur. "l" işareti veya anlamına gelir. Sondaki 1 değeri kuralın diğer kurallara göre öncelik sırasını belirtir. "SMS" değeri, bu kurala uyan sorguların ilgi olduğu servisi belirtmektedir. Sorgudan sonra gelen "NER_TEXT" kuralın tipidir, programa kendisinin nasıl işlenmesi gerektiğini gösterir. Mevcut kural tipleri aşağıda açıklanmıştır.



Şekil 4.4: Sınıflandırma Akışı

PLAIN_TEXT: Düz metinden oluşan kurallardır. Bu kurallarda cümlelerin DDI aşamaları ile işlenmiş halinden yararlanılmaz. Kullanıcı sorgusu hiçbir aşamadan geçirilmeden, kurala göre kontrol edilir.

NER_TEXT: Cümlelerde varlık ismi (NE) etiketi bulunan kelimeler, bu etiket ile değiştirilir. Kural buna göre kontrol edilir.

POS_TEXT: Cümlelerin POS etiketleri ile cümleler kontrol edilir.

POS_PLAIN_TEXT: Cümledeki bazı kelimelerin yerine POS etiketleri yazılır. Örneğin, "annemi ara" sorgusu bu tip için "<NN> ara" şeklinde yazılır.

Sorguların kurallara uyup uymadığını kontrol etmek oldukça hızlı bir işlemdir. Fakat her biri yalnızca özel durumları kapsamaktadır. Kapsadığı durum için çoğu kural yüzde yüz başarıya sahiptir; kesinliği (precision) yüksek, anması (recall) düşüktür. Performans açısından bir sorguyu kurallarla kontrol etmek çok hızlı ve doğrusal bir işlemdir. Performans konusunda istatistiksel yöntemlerden çok daha avantajlıdır ve eğitmek gerekmemektedir. Ancak, çok fazla kural eklendiğinde bakımı zorlaşır. Ayrıca her durumu karşılayacak kurallar yazmak da pratikte imkansızdır. Bu yüzden sadece genel durumlar için kurallar yazılmış ve yapı istatistiksel yöntem ile birleştirilmiştir. Sistemde toplam olarak 38 adet kural bulunmaktadır.

4.3.2 İstatistiksel yöntemler

Pratikte tüm durumları kapsayacak kadar kural tanımlayamayacağımız için istatistiksel bir yöntem kullanmamız gerekmiştir. Bu amaçla 1000 cümlelik bir eğitim verisi hazırlanmış ve bu veri birçok makine öğrenmesi algoritması kullanılarak test edilmiş ve en yüksek başarı veren algoritma uygulamada kullanılmak üzere seçilmiştir. Makine öğrenmesi algoritmaları geliştirme kapsamında yeniden yazılmamıştır, istatistiksel model oluşturmak ve yaklaşımların başarılarını ölçmek için Weka [26] aracı kullanılmıştır. Sınıflandırma işlemi için kullanılacak algoritmaya karar verilmeden birçok algoritma test edilmiş ve en başarılı sonucu veren **Destek Vektör Makineleri** (SVM) algoritması seçilmiştir. Kullanılan algoritmalar Bölüm 5.3.2'de açıklanmıştır. Ürettiğimiz veri yeterince büyük olmadığı için testler aynı veri üzerinde

10 kat çapraz doğrulama kullanılarak yapılmıştır (bkz. Bölüm 5.3). Verinin nasıl kullanıldığı ve hangi işlemlerden geçirildiği Bölüm 5.3.1’de detaylıca açıklanmıştır.

4.4 Parametre Çıkarımı

Sınıflandırma adımından sonra her bir servis için parametre çıkarım adımı çalışır. Bu adımda her sınıfa özgü algoritmalar tasarlanmıştır. Çizelge 4.2 sınıf bazında çıkartılması hedeflenen parametreleri göstermektedir. Parametre çıkarımı için önceki adımlarda oluşturulan biçimbilimsel analiz, varlık ismi tanıma ve bağlılık ayrıştırıcı araçlarının çıktıları kullanılmaktadır.

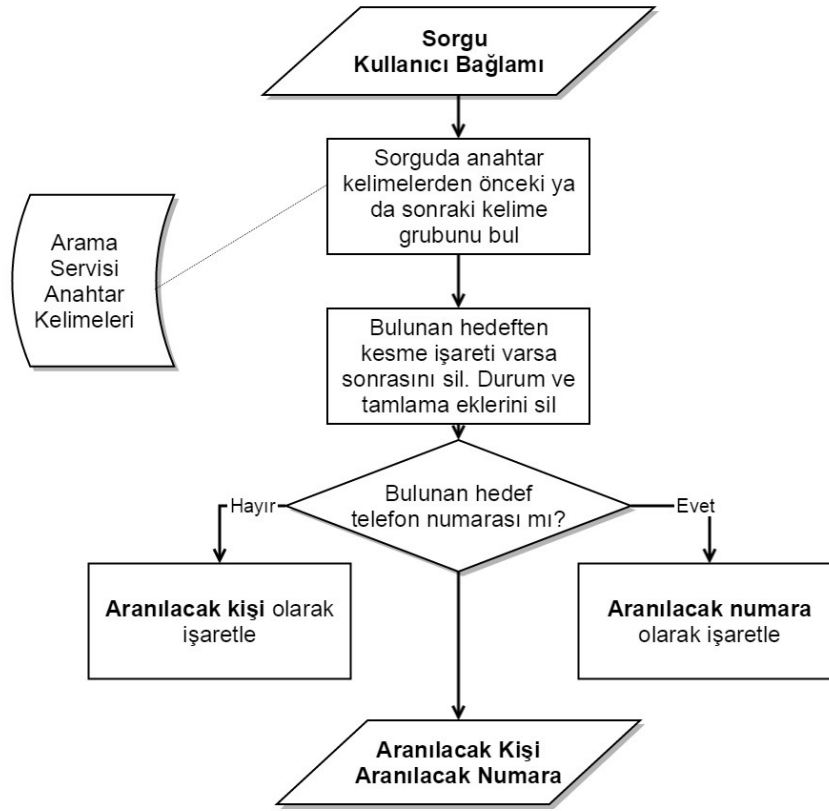
Çizelge 4.2: Sınıf bazında çıkartılan parametreler.

Sınıf	Parametreler
Arama	alıcı adı, telefon no
Mesaj	içerik, alıcı adı, telefon no
E-Posta	içerik, alıcı adı, e-posta adresi
İnternette Arama	sorgu, site
Uygulama	uygulama Adı
Harita	başlangıç, bitiş, hedef ilgi noktası mı
Hava Durumu	yer, zaman
Döviz	dönüşecek-dönüştürülecek kur, miktar
Trafik	

Her sınıfın (arama, mesaj ...) kendine özgü parametre çıkarıcısı bulunmaktadır. Sınıflandırma adımında belirlenen sınıfa göre ilgili ayrıştırıcı çalışır ve gerekli parametreleri belirler.

4.4.1 Arama

Kullanıcı sorgusunun arama servisi olduğu anlaşıldıktan sonra aranılacak kişi veya telefon numarası bulunmaya çalışılır. Bu sınıf için ara, çevir, çağrı gibi anahtar kelimeler tanımlanmıştır. Bu kelimelerden önceki veya sonraki kısım alıcı olarak belirlenir. "Lütfen", "rica ederim" gibi önemsiz kelimeler varsa çıkartılır. Alıcı isminden tamlayan ve yönelme hali ekleri silinir. Parametre çıkarımı algoritmasının akışı Şekil 4.5’te gösterilmiştir.



Şekil 4.5: Arama Servisi Parametre Çıkarım Akışı

4.4.2 Mesaj

Mesaj servisinde parametre çıkarımı için öncelikle "gönder, sms, mesaj" gibi bazı anahtar isimler, "gönder, at" gibi fiiller ve "lüften, rica etsem" gibi rica kelimeleri tanımlanmıştır. Program belirlenen tüm rica kelimelerini, isimleri ve fiilleri birleştirerek anahtar kelime listesi oluşturur. Ardından cümlede bu anahtar kelime gruplarını arar. Bulamaması durumunda belirlenen isimlerin çekimli olma durumunu bir düzenli ifade (regex) ile kontrol eder. Eşleşme olması durumunda, eşleşmeden önceki veya sonrasındaki kelimelere bakılır, yönelme halinde olan kelime grubu bulunursa, aldığı çekim ekleri silinir ve alıcı olarak işaretlenir. İşaretlenen kelime grubunun telefon numarası mı kişi ismi mi olduğu tespit edilir ve buna göre dönen yanıtta ilgili alan doldurulur. Sorgudan anahtar kelimeler ve alıcı bilgisi çıkarıldıktan sonra kalan kısım, mesajın içeriği olarak işaretlenir.

4.4.3 E-Posta

E-posta için kullanılan yöntem mesaj servisindeki ile aynıdır. Sadece anahtar isimler değiştirilmiştir.

4.4.4 Harita

Harita servisi için parametre çıkarımında başlangıç noktası, hedef bölge ve ilgi noktası çıkarılmaya çalışılır. Bu parametreleri tespit etmek için biçimbilimsel analiz ve bağlılık ayrıştırıcısı araçlarının çıktıları kullanılır. Bu modülün akışı Şekil 4.6'da gösterilmiştir.

4.4.5 Uygulama

"Çalıştır, başlat, aç, eriş, git ve güncelle" anahtar kelimeleriyle ilişkileri olan isim ve isim tamlamaları bulunur. Bu kelime grubundaki yönelme ekleri silinir ve uygulama ismi olarak kabul edilir. Eğer kullanıcı anahtar kelimelerden birini söylememişse, o zaman söylenen kelimeler uygulama ismi olarak kabul edilir "okeyi başlat" yerine sadece "okey" söylenmesi bu duruma örnek olarak verilebilir. Bu modülün akışı Şekil 4.7'de gösterilmiştir.

4.4.6 İnternette arama

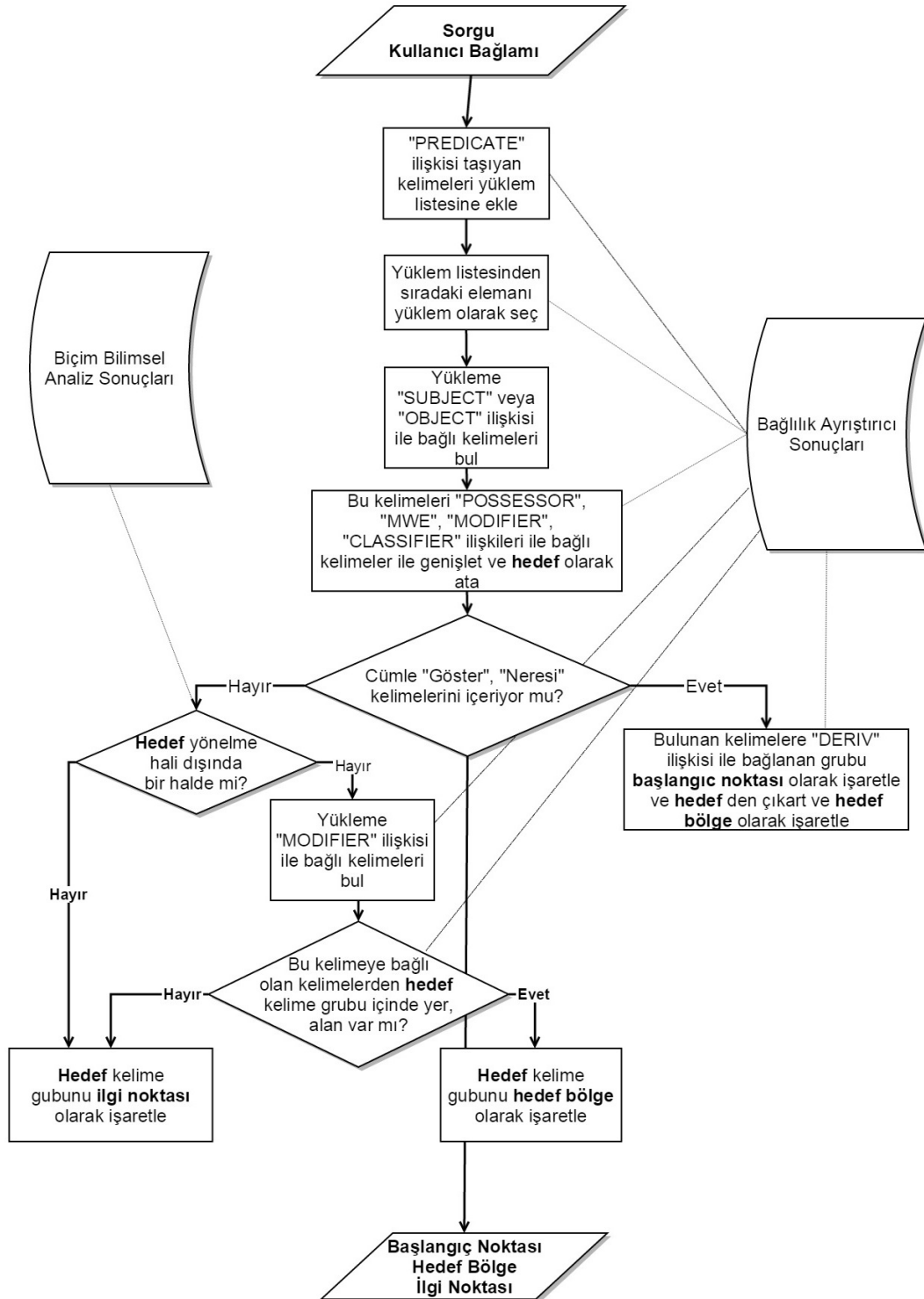
Kullanıcı sorgusunun internette arama servisi olduğu anlaşıldıktan sonra aranılacak metin ve site bulunmaya çalışılır. Bu sınıf için ara, git, bul gibi anahtar kelimeler tanımlanmıştır. Bu kelimelerden önceki veya sonraki kelime grubu aranacak metin olarak belirlenir. "Lütfen, rica ederim" gibi önemsiz kelimeler varsa çıkartılır. Aranılacak metinden tamlayan ve yönelme hali ekleri silinir. Parametre çıkarımı algoritmasının akışı Şekil 4.8'de gösterilmiştir.

4.4.7 Hava durumu

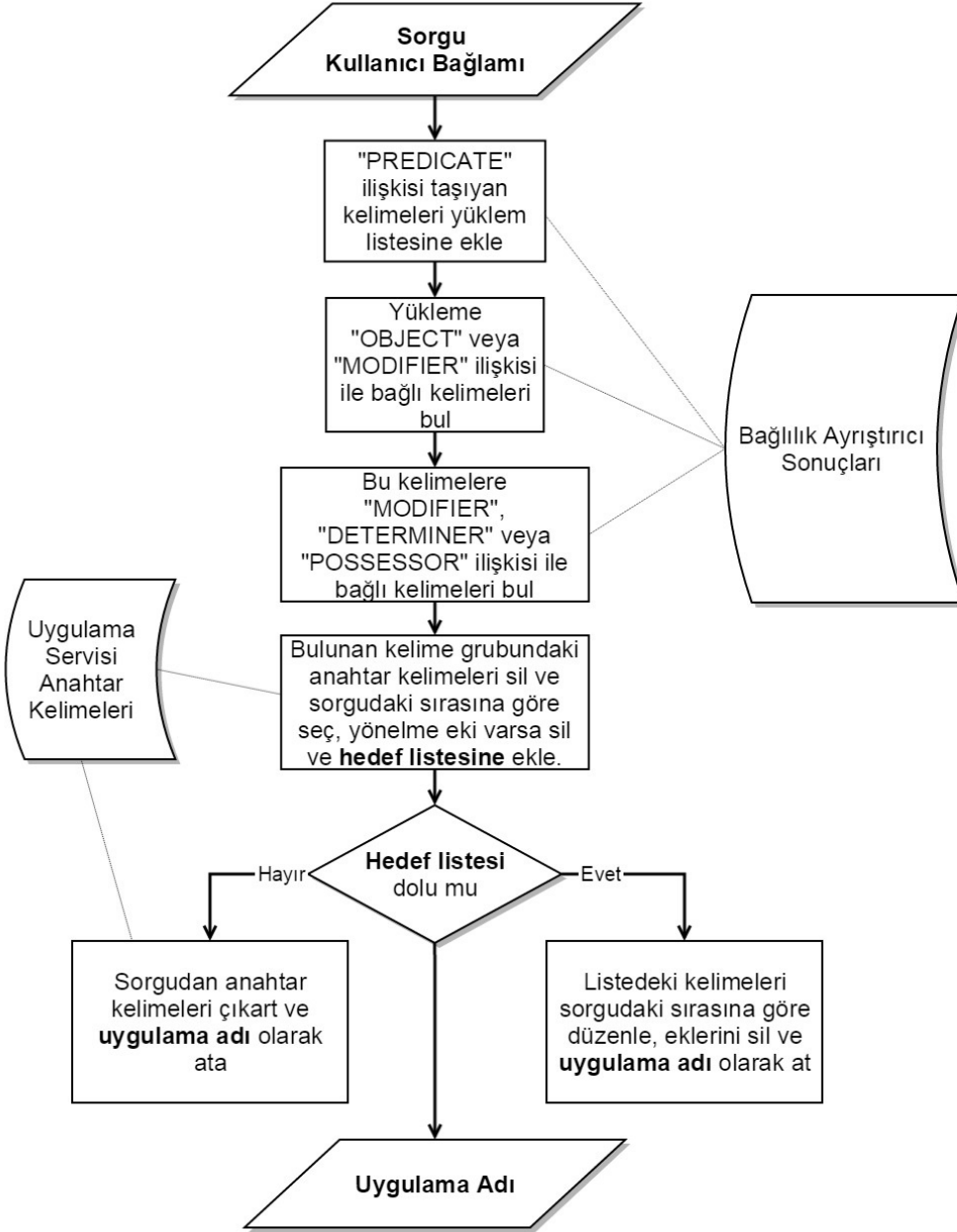
Kullanıcı sorgusundan konum ve zaman bilgileri çıkartılmaya çalışılır. Bu işlem yapılırken bağlılık ayrıştırıcısı, biçimbilimsel analiz ve varlık ismi tanıma araçlarının çıktıları kullanılır. Eğer sorguda bulunamadıysa bu değerler kullanıcı bağlamından alınır. Parametre çıkarımı algoritmasının akışı Şekil 4.9'da gösterilmiştir.

4.4.8 Döviz

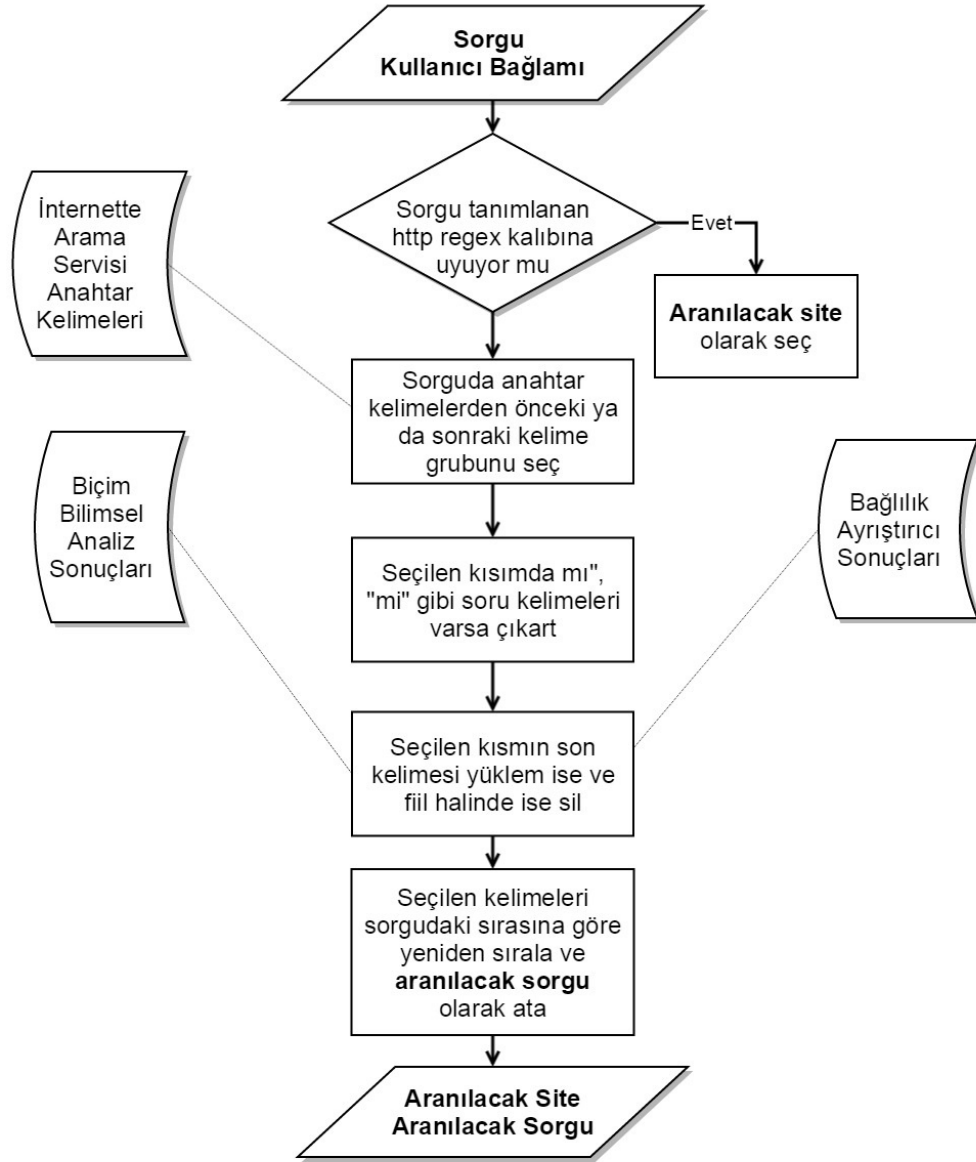
Döviz servisinde çevrilecek ve çevrilen para birimleri ve döviz miktarı parametreleri çıkartılmaya çalışılır. Para birimi sözlüğü, bağıklık ayrıştırıcısı ve biçimbilimsel analiz araçlarının çıktıları aktif bir şekilde kullanılır. Eğer bu DDİ araçlarının çıktıları kullanarak bir sonuç bulamazsa, doğrudan kullanıcı sorgusundaki kelimeleri tek tek para birimi sözlüğünde arar ve istekte geçiş sırasına göre çevrilen ve çevrilecek para birimleri olarak seçer. Eğer bu yöntemle de bulunamazsa varsayılan olarak çevrilen para birimine "Türk Lirası (TRY)", çevrilecek olana "Amerikan Doları (USD)" atanır ve miktar olarak da "1" seçilir. Bu modülün çıkışma akışı Şekil 4.10'da daha detaylı olarak gösterilmiştir.



Şekil 4.6: Harita Servisi Parametre Çıkarım Akışı



Şekil 4.7: Uygulama Servisi Parametre Çıkarım Akışı



Şekil 4.8: İnternette Arama Servisi Parametre Çıkarım Akışı

5. TEST SONUÇLARI

5.1 Veri

Tez kapsamında eğitim ve test amaçlı kullanılmak üzere veri toplanmış ve işaretlenmiştir. Bu kapsamda aşağıdaki işaretleme türlerinde çalışmalar yapılmıştır:

- Konuşmadan yazılı metne dönüştürülmüş veri (1000 cümle)
- Bu verinin elle normalize edilmiş hali
- Varlık ismi işaretleme (Kişi Adları, Yer Adları, Kurum Adları, Zaman, Para, Sayısal İfade) (750 cümle)
- Bağlılık ayrıştırıcısı İşaretleme (620 cümle)
- Mesajın hangi servise ait olduğunda dair işaretleme (1000 cümle)
- Mesajın ilgili servis ve alt alanlarının işaretlenmesi (578 beklenen sonuç)

Elimizdeki veri bilgisi Çizelge 5.1’de gösterilmiştir. Bu tablodaki veri tipi sütunu Çizelge 5.2’de açıklanmıştır.

Çizelge 5.1: Veri Bilgisi

Kullanım Amacı	Veri Tipi
Konuşmadan yazılı metne dönüştürülmüş veri	STT_Data
Bu verinin elle normalize edilmiş hali	STT_Data
NER Eğitimi (Kişi, Yer, Kurum, Zaman, Para, Sayısal İfade)	NER_Data
Sınıflandırma adımındaki istatistiksel yöntemlerin eğitimi	STT_Data
Parametre çıkarımı adımındaki başarının artırılması	Validation
Parametre çıkarımı adımındaki başarının ölçülmesi	Test

Çizelge 5.2: Veri Tipi Tanımı

Veri Adı	Cümle Sayısı	İşaretleme Tipi
STT_Data	1000	Normalize edilmiş ve sınıf işaretleri elle işaretlenmiş veri
NER_Data	750	Varlık isimleri işaretlenmiş veri
Parser_Data	520	Bağlılık ayrıştırıcı İşaretleme (STT_Data [439] ve Validation [81] kümelerinden derlenmiştir)
Validation	478	İlgili servis ve parametreleri işaretlenmiş veri
Test	100	İlgili servis ve parametreleri işaretlenmiş veri

5.2 Bağlılık Ayrıştırıcı Başarısı

Bağlılık ayrıştırıcı aracının eğitimi için 520 cümlelik veri (Parser_Data) işaretlenmiştir. Fakat bu veri 10’lu çapraz doğrulama ile kullanmak için yeterince büyük boyutta olmadığından 100 cümlelik test verisi de işaretlenerek bu kümeye eklenmiştir. 100 cümlelik veri sadece test adımı kullanılmış, sistemde kullanılan modelin eğitiminde bu veri çıkartılmıştır. Verinin tamamı İTÜ Annotation Tool [27] kullanılarak işaretlenmiştir. Öncelikle işaretlenen bu 620 cümlelik veri IMST (ITUMetuSabancıTreebank) [28] verisi ile eğitilmiş model üzerinde test edilmiştir. Bu testin sonuçları Çizelge 5.3’te gösterilmiştir.

Çizelge 5.3: IMST Verisi ile Eğitilmiş Bağlılık Ayrıştırıcı Modelinin Üretilen Veri Üzerindeki Başarımı

İşaretli Parça Puanı	27.55 %
İşaretsiz Parça Puanı	52.94 %
İşaret Doğruluk Puanı	34.14 %

Bu test sonucunda alan adaptasyonu gerekeceği ön görüldüğü için düşük bir başarı beklenmekteydi fakat bu kadar düşük beklenmiyordu. Bu düşüklüğün nedeni araştırıldı ve üretilen konuşma verisinin sonunda noktalama işareti bulunmamasından kaynaklandığı fark edildi. Ardından üretilen cümlelerin sonuna nokta eklenerek bu test tekrarlandı ve sonuçlarda önemli bir artış gözlemlendi (Bknz. Çizelge 5.4)

Çizelge 5.4: Üretilen Veriye Nokta Eklendikten Sonra IMST Verisi ile Eğitilmiş Bağlılık Ayrıştırıcı Modelinin Başarısı

İşaretli Parça Puanı	57.05 %
İşaretsiz Parça Puanı	76.79 %
İşaret Doğruluk Puanı	65.20 %

Çizelge 5.5: Üretilen Verinin 10-kat Çapraz Doğrulama Sonucu Ortalama Başarısı

İşaretli Parça Puanı	71.53 %
İşaretsiz Parça Puanı	86.58 %
İşaret Doğruluk Puanı	74.80 %

Bu test sonucunda resmi metin ile derlenen modeli kullanırken, cümlelerin sonunda noktalama işareti olmasının başarıyı artırdığı gözlenmiştir. Fakat çalışılan alan oldukça spesifik olduğundan bu işlem başarıyı yeterince iyi seviyelere çekememiştir. Bunun sonucunda alan adaptasyonunun gerektiğine karar verilmiştir.

Üretilen veriyi kendi içinde 10-kat çapraz doğrulama yöntemi ile test edildi ve çalışılan alan için Çizelge 5.4'te de gösterildiği üzere %10 civarında bir başarı artışı elde edilmiştir.

En yüksek sonuç bu alana özel üretilen veri ile alındığı için uygulamada bu verinin tamamı ile eğitilen model kullanılmıştır.

5.3 Sınıflandırma Başarısı

İstatistiksel sınıflandırıcıların eğitimi amacıyla 1000 cümlelik bir veri seti (STT_Data) hazırlanmıştır. Ayrıca bu setteki her bir kelime için gövdeleme işlemi uygulanmıştır ve varsa varlık isimleri etiketleri kelimenin kendisiyle değiştirilmiştir. Bu sayede aynı kökteki kelimeler, farklı ekler alsalar dahi aynı kelime gibi değerlendirilebilmiş ve "Gökhan, Gülşen, Ahmet" gibi farklı isimler <Person> etiketine dönüşerek farklılıkları giderilmiştir. Örnek olarak "erdinç i arar mısın lütfen" cümlesi "<PERSON> ara mı lütfen" cümlesine dönüştürülmüştür. Düzenlenmiş cümleyi ve ilgili olduğu sınıf bilgisini içeren veri Weka [26] kütüphanesinin anlayabileceği ".arff" formatına getirilip kaydedilmiştir. Şekil 5.1 hazırlanan verinin bir kısmını göstermektedir.

```
"<money> yüksel",doviz  
"<person> ara",arama  
"<person> mail at biraz geri gel",email  
"<location> <time> hava yağmurlu ol",hava_durumu  
"galeri aç",uygulama  
"<location> balıkçı göster",harita  
"<person> mesaj gönder.",sms  
"<money> kaç <money>",doviz  
"trafik durum göster",trafik
```

Şekil 5.1: Sınıflandırma Verisinin Bir Kısım

5.3.1 Verinin işlenmesi

Oluşturulan veri kümesi yalın hali ile makine öğrenmesi algoritmaları ile eğitim için kullanılamaz. Birçok algoritma sadece sayısal ve kategori olan veriler ile kullanılabilir. Oluşturulan veri kümesi ise düz metinden oluştuğu için *Bag of Words* [29] yöntemi ile kelimeler sayısal ifadelere dönüşmüştür. Bu yöntem ile veri setindeki sorgular kelime vektörleri ile ifade edilir hale getirilir. Vektördeki her indeks eşsiz bir kelimenin sorgudaki tekrar oranını temsil eder. Tüm vektörler ve sınıflandırmaları (el ile işaretlenmiş, sorgunun yönlendirilmesi gereken doğru servis) eğitim için gerekli olan veri setini oluşturur. Kökü aynı olup farklı ek almış kelimeler gövdeleyici sayesinde aynı kabul edilir. Böylece anlamsal olarak aynı olan kelimeler aynı indekste gruplandırılır. Sonuçta sadece düz metinlerden oluşan veri kümesi, vektörler haline getirilerek, üstünde algoritmaların çalıştırılacağı bir hale getirilmiş olur.

5.3.2 Kullanılan algoritmalar

Lojistik Regresyon [30] : Bu yöntemde eğitim örneklerine en yakın doğru oluşturulmaya çalışılır. Bu doğru eşik olarak belirlenir. Düzlem doğru tarafından iki bölüme ayrılır. Örnekler, düzlemde doğrunun hangi tarafında kaldığına bağlı olarak pozitif veya negatif şeklinde işaretlenir. Bu algoritma sadece iki sınıf bulunan durumlar için kullanılır. Projemizde ikiden fazla sınıf bulunduğu için çoklu sınıf sınıflandırıcıları kullanılmıştır. Çoklu sınıf sınıflandırıcı türü olarak bire bir seçilmiştir. Bu yaklaşıma göre gelen örnek sınıflandırılırken her sınıf diğer sınıflarla tek tek karşılaştırılır ve diğerlerine üstün gelen sınıf seçilir.

Destek vektör makineleri (SVM) [31, 32] : Sınıflandırma konusunda kullanılan oldukça etkili ve basit yöntemlerden birisidir. Bir düzlemde bulunan iki grup arasında bir sınır çizilerek iki grubu ayırmak mümkündür. Bu sınırın çizileceği yer ise iki grubun da üyelerine en uzak olan yer olmalıdır. Bu işlemin yapılması için iki gruba da yakın ve birbirine paralel iki sınır çizgisi çizilir ve bu sınır çizgileri birbirine yaklaştırılarak ortak sınır çizgisi üretilir. Bu yöntem özellikle metin sınıflandırma konusunda çok popüler bir algoritmadır.

Naive Bayes [33] : Naive Bayes temelini Bayes teoreminden alır. Eğitim örneklerine bakılarak ve Bayes teoremi uygulanarak, kelimelerin veya kelime gruplarının her sınıfta kaç kez tekrarlandığına bakılır. Yeni bir örnek geldiğinde bu cümledeki kelimelere göre her bir sınıfa ait olma olasılığı hesaplanır ve en yüksek olasılık hesaplanan sınıf seçilir. Naive Bayes, sınıflandırma algoritmaları arasında en basit yöntemlerden birisidir ve çok yaygın bir biçimde kullanılır. Sonuçlar kısmında gösterilen Multinomial Bayes ve Bayes Net, Naive Bayes algoritmasının türevleridir.

Karar Ağaçları [34] : Bu yöntemde, bir ağaç yapısı oluşturulur. Ağacın her dalı, eğitim verisindeki bir özelliğe göre ağacı yapraklara ayırır. Özellikler, veri kümesini sayıları birbirine yakın gruplara bölecek şekilde seçilir. Daha fazla bölünemiyorsa, yani yapraktaki tüm örnekler tek bir sınıfa ait ise bu yaprağa gelen örnekler bu sınıf olarak işaretlenir. Karar ağaçları değişik *güvenirlilik faktörü* değerleri ile eğitilmiştir. Bu değerler bir yapraktaki sınıfların dominant sınıfa oranını temsil eder. Eğer oran belirlenen değerden düşük ise, yaprak daha fazla dallandırılmaz. Projemizde güvenlik faktörü 1, 0.5, ve 0.25 olarak test edilmiştir.

K-en yakın komşu [35] : Bu algoritmada eğitim verisinde her sınıfa ait olan örneklerin orta noktası çıkarılır. Yeni bir örnek gelince, bir uzaklık fonksiyonu ile bu orta noktalara uzaklığı hesaplanır ve en yakın olan orta noktanın ait olduğu sınıf etiketlenir.

5.3.3 Sınıflandırma testleri

Makine öğrenme algoritmalarının bir çoğu sadece ikili sınıflandırma yapabilmektedir. Ancak bizim durumumuzda toplam 9 tane servis olduğu için, sınıflandırıcılar bu sınıflar arasında bir seçim yapmalıdır. Bazı algoritmalar, örnek olarak; karar ağaçları ve Naive Bayes; bu durum için uygundur. Fakat, lojistik regresyon gibi algoritmalar

bu beceriye sahip değildir. Bu yüzden model oluşturma işleminde çoklu sınıf sınıflandırıcıları kullanılmıştır. Çoklu sınıf sınıflandırıcı türü olarak 1 vs 1 seçilmiştir.

Çizelge 5.6: Gövdeleme ve NER adımlarından Önce ve Sonra Sınıflandırma Başarısı

{Algoritma}	Gövdeleme ve NER adımlarından Önce		Gövdeleme ve NER adımlarından Sonra	
Algoritma	Doğruluk	Fskoru	Doğruluk	Fskoru
<i>Karar Ağacı 0.25</i>	85,90%	86.1%	89,90%	90.0%
<i>Karar Ağacı 0.5</i>	85,70%	86.0%	90,30%	90.4%
<i>Karar Ağacı 1</i>	85,40%	85.8%	90,30%	90.4%
<i>Lojistik Regresyon</i>	94,10%	94.1%	95,20%	95.2%
<i>NaiveBayes Multinomial</i>	92,60%	92.2%	93,80%	93.5%
<i>NaiveBayes</i>	92,20%	92.2%	93,80%	93.8%
<i>Bayes Net</i>	82,50%	83.0%	92,30%	92.3%
<i>K-En Yakın Komşu</i>	87,90%	87.8%	90,40%	90.4%
<i>SVM</i>	95,30%	95.5%	95,80%	95.8%

Çizelge 5.7: Bayes Net Algoritmasının Sınıf Bazında Başarımı

Sınıf	Doğruluk	F Skoru
<i>arama</i>	97,00%	91.5%
<i>doviz</i>	92,00%	94.8%
<i>email</i>	97,00%	93.3%
<i>harita</i>	84,80%	87.8%
<i>hava durumu</i>	97,80%	98.6%
<i>mesaj</i>	93,60%	95.1%
<i>trafik</i>	85,70%	92.3%
<i>uygulama</i>	91,90%	87.1%
<i>internette arama</i>	88,60%	92.2%
Ortalama	92,30%	92.3%

Veri kümesi küçük olduğu için 10'lu çapraz doğrulama uygulanmış ve şans faktörünü ortadan kaldırmak için bu işlem 10 kez tekrar edilmiştir. Sonuçlar bu 10 test koşusunun ortalamalarını göstermektedir. Kelimelerde çekim eklerinin ve varlık isimlerinin test başarısını düşüreceği öngörüldüğü için testler gövdeleme ve NER adımlarından öncesi ve sonrası şeklinde yapılmıştır. Çizelge 5.6'da görüldüğü üzere doğal dil işleme araçlarını kullanmak, sınıflandırma adımıyla test edilen tüm algoritmalarda başarıyı artırmıştır. Karar ağaçları ve k en yakın komşu algoritmalarının bu araçlardan kullanımından en çok etkilenen algoritmalar olduğu gözlenmiştir.

Çizelge 5.8: K-En Yakın Komşu Algoritmasının Sınıf Bazında Başarımı

Sınıf	Doğruluk	F Skoru
<i>arama</i>	99,00%	94.7%
<i>doviz</i>	93,00%	92.5%
<i>email</i>	78,00%	83.9%
<i>harita</i>	86,10%	88.5%
<i>hava durumu</i>	97,80%	97.8%
<i>mesaj</i>	92,00%	90.2%
<i>trafik</i>	90,50%	88.4%
<i>uygulama</i>	89,70%	84.7%
<i>internette arama</i>	87,70%	91.3%
Ortalama	90,40%	90.4%

Çizelge 5.9: 0.25 Güvenirlik Faktörlü Karar Ağacının Sınıf Bazında Başarımı

Sınıf	Doğruluk	F Skoru
<i>arama</i>	95,00%	93.6%
<i>doviz</i>	85,00%	85.4%
<i>email</i>	86,00%	90.1%
<i>harita</i>	87,30%	87.8%
<i>hava durumu</i>	97,80%	96.8%
<i>mesaj</i>	96,00%	94.9%
<i>trafik</i>	85,70%	92.3%
<i>uygulama</i>	83,10%	79.3%
<i>internette arama</i>	89,50%	92.3%
Ortalama	89,90%	90.0%

Çizelge 5.10: 0.50 Güvenirlik Faktörlü Karar Ağacının Sınıf Bazında Başarımı

Sınıf	Doğruluk	F Skoru
<i>arama</i>	96,00%	92.8%
<i>doviz</i>	86,00%	86.9%
<i>email</i>	88,00%	91.2%
<i>harita</i>	87,30%	89.2%
<i>hava durumu</i>	97,80%	96.8%
<i>mesaj</i>	96,00%	94.9%
<i>trafik</i>	85,70%	92.3%
<i>uygulama</i>	83,10%	79.0%
<i>internette arama</i>	89,50%	92.7%
Ortalama	90,30%	90.4%

Çizelge 5.11: 1 Güvenirlik Faktörlü Karar Ağacının Sınıf Bazında Başarımı

Sınıf	Doğruluk	F Skoru
<i>arama</i>	95,00%	93.6%
<i>doviz</i>	88,00%	87.6%
<i>email</i>	86,00%	87.3%
<i>harita</i>	84,20%	88.3%
<i>hava durumu</i>	97,80%	97.1%
<i>mesaj</i>	95,20%	95.2%
<i>trafik</i>	85,70%	92.3%
<i>uygulama</i>	88,20%	81.4%
<i>internette arama</i>	89,50%	92.7%
Ortalama	90,30%	90.4%

Çizelge 5.12: Lojistik Regresyon Algoritmasının Sınıf Bazında Başarımı

Sınıf	Doğruluk	F Skoru
<i>arama</i>	99,00%	94.3%
<i>doviz</i>	98,00%	97.5%
<i>email</i>	93,00%	92.5%
<i>harita</i>	96,40%	95.5%
<i>hava durumu</i>	100,00%	99.6%
<i>mesaj</i>	89,60%	92.6%
<i>trafik</i>	90,50%	92.7%
<i>uygulama</i>	95,60%	94.2%
<i>internette arama</i>	90,40%	94.9%
Ortalama	95,20%	95.2%

Çizelge 5.13: Naive Bayes Algoritmasının Sınıf Bazında Başarımı

Sınıf	Doğruluk	F Skoru
<i>arama</i>	96,00%	92.8%
<i>doviz</i>	94,00%	96.4%
<i>email</i>	99,00%	96.1%
<i>harita</i>	92,10%	91.0%
<i>hava durumu</i>	98,60%	98.6%
<i>mesaj</i>	92,80%	95.1%
<i>trafik</i>	90,50%	92.7%
<i>uygulama</i>	89,70%	90.0%
<i>internette arama</i>	90,40%	92.0%
Ortalama	93,80%	93.8%

Çizelge 5.14: Naive Bayes Multinomial Algoritmasının Sınıf Bazında Başarımı

Sınıf	Doğruluk	F Skoru
<i>arama</i>	97,00%	95.6%
<i>doviz</i>	92,00%	95.3%
<i>email</i>	96,00%	97.5%
<i>harita</i>	98,20%	92.8%
<i>hava durumu</i>	100,00%	98.9%
<i>mesaj</i>	96,00%	90.9%
<i>trafik</i>	33,30%	50.0%
<i>uygulama</i>	87,50%	92.2%
<i>internette arama</i>	93,00%	93.4%
Ortalama	93,80%	93.5%

Çizelge 5.15: SVM Algoritmasının Sınıf Bazında Başarımı

Sınıf	Doğruluk	F Skoru
<i>arama</i>	99,00%	95.7%
<i>doviz</i>	97,00%	97.0%
<i>email</i>	94,00%	96.9%
<i>harita</i>	93,90%	95.1%
<i>hava durumu</i>	100,00%	98.6%
<i>mesaj</i>	96,80%	96.8%
<i>trafik</i>	100,00%	100%
<i>uygulama</i>	93,40%	92.4%
<i>internette arama</i>	92,10%	93.8%
Ortalama	95,80%	95.8%

Çizelge 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15 gövdeleme ve NER adımlarından sonra test edilen algoritmaların sınıf bazında başarılarını göstermektedir. Test sonuçlarında görüldüğü üzere en yüksek başarı SVM algoritması kullanılarak elde edilmiştir.

İstatistiksel yöntem ile SVM algoritmasını kullanarak %95.80'lik başarıya ulaşılmıştır. Daha sonra istatistiksel yöntem melez bir yapı altında kural tabanlı yöntem ile birleştirilmiştir ve başarı daha da artırılmıştır. Geliştirilen melez yaklaşım ile tüm sistemin sınıflandırma başarısı **%98.30** olarak ölçülmüştür.

5.4 Parametre Çıkarımı Başarısı

Parametre çıkarımı adımında her sınıfa özel algoritma geliştirilmiştir. Bu algoritmalar biçim bilimsel analiz, varlık ismi tanıma ve bağıllık ayrıştırıcısı araçlarının çıktılarını bir arada kullanmaktadır.

5.4.1 Verinin hazırlanması

Sunucu tarafındaki parametre çıkarımı adımının başarısının ölçülmesi ve geliştirilmesi için farklı kişilerden 578 cümlelik veri (Validation + Test) toplanmış işaretlenmiştir. Bu verinin 478 cümlesi (Validation) sistemin başarısını artırmak amacıyla doğrulama verisi olarak kullanılmış, kalan 100 (Test) cümle başarısını ölçmek için kullanılmıştır. Bu bölümde belirtilen sonuçlar bu 100 cümle üzerinde elde edilmiştir. Verinin tamamı Google'ın konuşmadan metine dönüşüm servisi [12] kullanılarak toplanmış, cümle ve karşılığında gelmesi beklenen sonuç şeklinde işaretlenmiştir. İşaretlenen verinin bir kısmı **EK A.1**'de örnek olarak verilmiştir.

5.4.2 Parametre çıkarımı testleri

Geliştirilen sistem işaretlenen 100 cümlelik veri ile test edilmiştir. Ayrıca Turkcell Mobil Asistan, Asistan B, CEYD-A ve Siri uygulamaları da aynı veri test edilmiştir. Çizelge 5.16 test yapılan tüm uygulamaların sınıf bazındaki başarılarını göstermektedir. Diğer uygulamalar test edilirken sadece bu tez kapsamında geliştirilen uygulamanın desteklediği alanlardaki başarıları ölçülmüş, sohbet, alarm kurma

gibi diğer servislerdeki performansları değerlendirilmemiştir. Çizelgede "N/A" ile belirtilen alanlar uygulamanın bu servisi desteklemediğini göstermektedir.

Çizelge 5.16: Test Edilen Uygulamaların Sınıf Bazında Başarısı

Sınıf /Uygulama	AsistanB	Turkcell	CEYD-A	SIRI	Tez
Hava Durumu	0,00%	50,00%	50,00%	83,33%	91,67%
İnternette Arama	0,00%	0,00%	27,27%	63,64%	36,36%
Döviz	60,00%	30,00%	40,00%	N/A	90,00%
E-Posta	N/A	N/A	0,00%	36,36%	63,64%
Harita	0,00%	16,67%	66,67%	66,67%	75,00%
Arama	0,00%	0,00%	27,27%	63,64%	81,82%
Uygulama	50,00%	N/A	58,33%	91,67%	83,33%
Mesaj	0,00%	50,00%	10,00%	70,00%	40,00%
Ortalama	17,95%	34,85%	40,45%	69,62%	70,79%

Yapılan testlerin sonucunda kural tabanlı bir yaklaşım ile çalışan Turkcell Mobil Asistan, AsistanB ve CEYD-A uygulamalarının oldukça yetersiz olduğu gözlenmiştir. Bu problemin temel kaynağı, bu uygulamaların doğal dil işleme bileşenlerini kullanmaması ve soruları önceden belirlenmiş bir şablona oturtmaya çalışması olarak görülmüştür. Bu uygulamalar içerisinde CEYD-A uygulaması daha zengin bir kural yapısına sahiptir ve kullanıcıların kendi kurallarını tanımlamasına izin vermektedir. Hatta bu kuralların kolayca girilebilmesi ve işlenmesi için CEYD adı verilen bir dil geliştirilmiştir. Bu sebeplerden ötürü CEYD-A uygulaması diğer kural tabanlı uygulamalara kıyasla daha iyi bir performans göstermiştir.

Diğer yandan SIRI uygulaması ve bu tez kapsamında geliştirilen uygulama doğal dil işleme bileşenlerini aktif olarak kullandığından, farklı yapılarıdaki sorulara da yanıt verebilmektedir. Bu yetenek, test sonuçlarında da görüldüğü üzere kural tabanlı uygulamalara kıyasla daha başarılı olmalarını sağlamıştır.

SIRI ve geliştirilen uygulama test verisi üzerinde kıyaslandığında; internette arama, uygulama çalıştırma ve mesaj gönderme alanlarında SIRI'nin, hava durumu, e-posta gönderme, haritada arama ve telefonla arama sınıflarında geliştirilen uygulamanın daha iyi sonuç verdiği görülmüştür. Ayrıca döviz ile ilgili sorulara SIRI'nin destek vermediği görülmüştür.

"Atatürk'ün doğum tarihi kaç" gibi bilgi sorularında, SIRI'nin ulaşılmak istenen bilgiyi internette doğru şekilde aradığı ve bu alanda geliştirilen uygulamadan daha iyi ve net yanıt döndürdüğü tespit edilmiştir. Mesaj gönderme alanında SIRI'nin daha esnek yapıdaki soruları desteklediği görülmüştür. Geliştirilen uygulama "gökhana sor nerdesin" benzeri sorularda "sormak" filini anlamadığı için yanıt veremezken SIRI bu kelimenin mesaj göndermek ile ilgili olduğunu düşünerek başarılı yanıt döndürmektedir. SIRI, mesaj alanında daha başarılı olduğu halde benzer bir servis olan e-posta gönderiminde ilgili parametrelerin çıkarımı konusunda oldukça yetersiz kalmıştır. "Gökhan'a daha sonra geri döneceğimi email at" benzeri sorularda geliştirilen uygulama başarılı bir şekilde alıcı ve e-posta içeriğini çıkarttığı halde SIRI bu parametreleri çıkartamamış kime gönderilecek, içeriği ne olacak gibi sorularda bu parametreleri tekrar sormuştur. Telefon rehberinde kayıtlı kişileri arama, haritada yol tarifi alanlarında da SIRI'nin parametre çıkartma konusunda yetersiz kaldığı görülmüştür.

6. SONUÇ VE ÖNERİLER

Günümüzde bilgiye erişimi hızlandırmak ve kolaylaştırmak temel araştırma alanlarından birisi olmuştur. Bu amaçla geliştirilen mobil sesli asistan uygulamaları da oldukça dikkat çekmiş ve ciddi bir kullanıma ulaşmıştır. Siri ve Google Now en başarılı ve yaygın uygulamalar olsa da birçok farklı uygulama da marketlerde yer almaktadır. Siri uygulaması IOS 8.3 güncellemesinden sonra Türkçe desteği sunmaya başlamış olsa da İngilizce için sunduğu servis ile aynı kalitede değildir. Ayrıca diğer uygulamaların neredeyse hiçbirinin Türkçe desteği bulunmamaktadır.

Türkçe için geliştirilen uygulamalar da bulunmaktadır fakat yapılan incelemeler sonucunda, yeterli doğal dil işleme yöntemleri kullanmadığı ve başarılarının düşük olduğu gözlenmiştir. Bu ihtiyaç doğrultusunda tez kapsamında Türkçe için yapılmış olan doğal dil işleme çalışmalarından da faydalanacak bir yapı tasarlanmıştır.

Geliştirilen yapıda mobil cihaz ve sunucu olarak iki ayrı sistem tasarlanmış ve sorumluluklar dağıtılmıştır. Mobil cihaz kullanıcı ile haberleşme kısmından, sunucu tarafındaki yazılım da isteğin anlaşılmasından sorumlu olmuştur. İsteğin anlama kısmı da yine 2 adımda gerçekleşmiştir. Öncelikle gelen isteğin hangi konuda olduğu tespit edilmiş ardından sorgudan gerekli bilgiler çıkartılmış ve işlem yapılmıştır.

Gelen isteğin hangi konu ile ilgili olduğu anlama adımında melez bir yaklaşım geliştirilmiştir. Kural motoru geliştirilmiş ve kurallar tanımlanmıştır, kurallara uyulmama durumunda da istatistiksel bir model geliştirilmiştir. Bu modelin geliştirilmesinde birçok algoritma denenmiş, en iyi sonuç SVM algoritması ile %95.8 olarak elde edilmiştir. İstatistiksel yöntem ile kural tabanlı yöntem birleştirilince başarı daha da artmış ve %98.3 olarak hesaplanmıştır. Parametre çıkarımı adımında her servis için farklı bir algoritma gelişmiş ve bunların detayları tezde verilmiştir. Sistemin başarısı sınıf bazında değişmekte olup ortalama %70.23 olarak ölçülmüştür.

Uygulamada kullanıcı tarafından sorulan sorgular kayıt edilmektedir. İleriki zamanlarda kullanıcılardan geri dönüş alınarak sorgular işaretlenebilir ve aktif öğrenme algoritması geliştirilerek kullanılabilir. Bunun dışında yanıtları sesli söyleme ve kullanıcı ile diyalog yapıları kurularak kullanıcı deneyimi alanında iyileştirmeler yapılabilir.

KAYNAKLAR

- [1] Apple - iOS - Siri, <http://www.apple.com/ios/siri/>.
- [2] 2015, Google, <https://play.google.com/store/apps/details?id=com.google.android.googlequicksearchbox>.
- [3] **Neustein, A. ve Markowitz, J.A.**, 2013. Mobile Speech and Advanced Natural Language Solutions, Springer Science & Business Media, <https://www.google.com/books?hl=tr&lr=&id=TWZDAAAAQBAJ&oi=fnd&pg=PR3&dq=Mobile+Speech+and+Advanced+Natural+Language+Solutions&ots=ttPtFoImW2&sig=PeU72DUiFoYKHHDQa-9rTj7SZQQ>.
- [4] **Hawkins, J.C., Bridgwater, T.B., Haitani, R.Y. ve Rees, W.B.**, 2010. Integrated personal digital assistant device, Google Patents, <https://www.google.com/patents/US7650147>, uS Patent 7,650,147.
- [5] **Eryigit, G.**, 2014. ITU Turkish NLP web service, *EACL 2014*, 1, <http://www.aclweb.org/anthology/E/E14/E14-2.pdf#page=15>.
- [6] **Çelikkaya, G., Torunoğlu, D. ve Eryiğit, G.**, 2013. Named Entity Recognition on Real Data: A Preliminary Investigation for Turkish., Proceedings of the 7th International Conference on Application of Information and Communication Technologies, AICT2013, IEEE, Baku, Azarbeijan.
- [7] **Çelikkaya, G. ve Eryigit, G.**, 2014. A Mobile Assistant for Turkish, Proceedings of the TURKLANG'14 International Conference on Turkic Language Processing, Istanbul, Turkey.
- [8] 2015, CEYD-A Türkçe Sesli Asistan, <https://play.google.com/store/apps/details?id=com.cenker.yardimci.app>.
- [9] 2015, Turkcell Mobil Asistan, <https://play.google.com/store/apps/details?id=com.turkcellteknoloji.MobilAsistan>.
- [10] 2015, Asistan B - Türkçe Sesli Asist, <https://play.google.com/store/apps/details?id=com.buronya.asistanb>.
- [11] Android'de Türkçe Anlayan Konuşan Türk Asistan İDRİS - ÇIKTI, <http://beta.donanimhaber.com/tm.asp?m=65818892>.

- [12] **Ballinger, B., Allauzen, C., Gruenstein, A. ve Schalkwyk, J.**, 2010. On-demand language model interpolation for mobile speech input., *INTERSPEECH*, s.1812–1815.
- [13] **Balani, N. ve Hathi, R.**, 2009. Apache Cxf web service development: Develop and deploy SOAP and RESTful web services, Packt Publishing Ltd.
- [14] **Monge, A. ve Elkan, C.**, 1996. The field matching problem: Algorithms and applications, In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, s.267–270.
- [15] **Yujian, L. ve Bo, L.**, 2007. A normalized Levenshtein distance metric, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **29(6)**, 1091–1095.
- [16] **Eryiğit, G., Çetin, F.S., Yanik, M., Temel, T. ve Çiçekli, I.**, 2013. TURKSENT: A Sentiment Annotation Tool for Social Media, *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, Association for Computational Linguistics, Sofia, Bulgaria*, s.131–134, <http://www.aclweb.org/anthology/W13-2316>.
- [17] **Şahin, M., Sulubacak, U. ve Eryiğit, G.**, 2013. Redefinition Of Turkish Morphology Using Flag Diacritics, *Proceedings of The Tenth Symposium on Natural Language Processing (SNLP-2013)*, Phuket, Thailand.
- [18] **Şeker, G.A. ve Eryiğit, G.**, 2012. Initial Explorations on using CRFs for Turkish Named Entity Recognition, *Proceedings of COLING 2012, Mumbai, India*.
- [19] **Eryiğit, G., Nivre, J. ve Oflazer, K.**, 2008. Dependency parsing of Turkish, *Computational Linguistics*, **34(3)**, 357–389.
- [20] **Lafferty, J., McCallum, A. ve Pereira, F.C.**, 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- [21] **Rabiner, L. ve Juang, B.H.**, 1986. An introduction to hidden Markov models, *ASSP Magazine, IEEE*, **3(1)**, 4–16.
- [22] **Kudo, T.**, 2014. CRF++: Yet another CRF toolkit (2005), *Software available at* <http://crfpp.sourceforge.net>.
- [23] **Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryiğit, G., Kübler, S., Marinov, S. ve Marsi, E.**, 2007. MaltParser: A Language-Independent System for Data-Driven Dependency Parsing, *Natural Language Engineering Journal*, **13(2)**, 99–135.
- [24] **Morante, R.**, 2008. Semantic Role Labeling Tools Trained on the Cast3LB-CoNNL-SemRol Corpus., *LREC*.
- [25] Regular expression, http://en.wikipedia.org/w/index.php?title=Regular_expression&oldid=654317915.

- [26] **Holmes, G., Donkin, A. ve Witten, I.H.**, 1994. Weka: A machine learning workbench, *Intelligent Information Systems*, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on, IEEE, s.357–361.
- [27] **Eryigit, G., Cetin, F.S., Yanık, M., Bilgi, T.G., Temel, T. ve Cicekli, I.**, 2013. Turksent: A sentiment annotation tool for social media, *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, Association for Computational Linguistics, Sofia, Bulgaria, s.131–134, <http://www.aclweb.org/anthology/W13-2316>.
- [28] **Sulubacak, U. ve Eryiğit, G.**, 2013. Representation of Morphosyntactic Units and Coordination Structures in the Turkish Dependency Treebank., *Proceedings of SPMRL 2013 (4th Workshop on Statistical Parsing of Morphologically Rich Languages)*, EMNLP 2013, Seattle, USA.
- [29] **Uijlings, J., Smeulders, A. ve Scha, R.**, 2009. Real-time Bag-of-Words, Approximately, CIVR, <http://www.huppelen.nl/publications/realTimeBoW.pdf>.
- [30] **le Cessie, S. ve van Houwelingen, J.**, 1992. Ridge Estimators in Logistic Regression, *Applied Statistics*, **41**(1), 191–201.
- [31] **EL-Manzalawy, Y.**, 2005, WLSVM, <http://www.cs.iastate.edu/~yasser/wlsvm/>, you don't need to include the WLSVM package in the CLASSPATH.
- [32] **Chang, C.C. ve Lin, C.J.**, 2001, LIBSVM - A Library for Support Vector Machines, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>, the Weka classifier works with version 2.82 of LIBSVM.
- [33] **John, G.H. ve Langley, P.**, 1995. Estimating Continuous Distributions in Bayesian Classifiers, *Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Mateo, s.338–345.
- [34] **Quinlan, R.**, 1993. C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA.
- [35] **Aha, D. ve Kibler, D.**, 1991. Instance-based learning algorithms, *Machine Learning*, **6**, 37–66.

EKLER

EK A.1 : Parametre Çıkarımı Verisi

EK A.1

```
req0001_1.request = halamı ara
req0001_2.expected =
{
  "responseCode": "api/status/ok",
  "responseType": "native_op",
  "nativeResponse": {
    "operationType": "native/call",
    "call": {
      "receiverName": "halam"
    }
  }
}
req0002_1.request = engine mesaj at yoldayım geliyorum
req0002_2.expected =
{
  "responseCode": "api/status/ok",
  "responseType": "native_op",
  "nativeResponse": {
    "operationType": "native/sms",
    "sms": {
      "receiverName": "engin",
      "bodyText": "yoldayım geliyorum"
    }
  }
}
req0003_1.request = twitter açar mısın
req0003_2.expected =
{
  "responseCode": "api/status/ok",
  "responseType": "native_op",
  "nativeResponse": {
    "operationType": "native/launch",
    "launch": {
      "appName": "twitter"
    }
  }
}
req0004_1.request = beylikdüzüne götür beni
req0004_2.expected =
{
```

```

    "responsecode": "api/status/ok",
    "responsetype": "native_op",
    "nativereponse": {
      "operationtype": "native/map",
      "map": {
        "routetarget": "beylikdüzü",
        "istargetlocality": false
      }
    }
  }
}
req0005_1.request = trafik nasıl
req0005_2.expected=
{
  "responseCode": "api/status/ok",
  "responseType": "native_op",
  "nativeResponse": {
    "operationType": "native/traffic"
  }
}
req0006_1.request = edirnedeki hava durumu
req0006_2.expected =
{
  "responseCode": "api/status/ok",
  "responseType": "service",
  "service": {
    "domainType": "weather",
    "images": [
      resim listesi
    ],
    "results": [
      sonuc listesi
    ]
  }
}
req0007_1.request = 4435 euro kaç dolar
req0007_2.expected =
{
  "responseCode": "api/status/ok",
  "responseType": "service",
  "service": {
    "domainType": "exchange",
    "results": [
      {
        "text": "Sonuç:",
        "subText": "4698,88"
      },
      {

```

```
    "text": "Döviz kuru:",  
    "subText": "1,06"  
  },  
  {  
    "text": "Çevirilecek para birimi:",  
    "subText": "EUR"  
  },  
  {  
    "text": "Çevirilen para birimi:",  
    "subText": "USD"  
  },  
  {  
    "text": "İstenilen miktar:",  
    "subText": "4435 yuro"  
  }  
]  
}
```




ÖZGEÇMİŞ

Ad Soyad: Gökhan Çelikkaya

Doğum Yeri ve Tarihi: İstanbul 15.01.1989

Adres: Osmaniye Mah. Fabrikalar Sok. No:6 Özyuva Apt. A Blok D:13 Bakırköy
İstanbul

E-Posta: gokhancelikkaya@gmail.com

Lisans: Bilgisayar Mühendisliği - Yıldız Teknik Üniversitesi

Y. Lisans: Bilgisayar Mühendisliği - İstanbul Teknik Üniversitesi

Mesleki Deneyim ve Ödüller:

03.2015 -	Amadeus	Kıdemli Yazılım Mühendisi
08.2014 - 03.2015	Sahibinden.com	Yazılım Mühendisi
08.2011 - 08.2014	Huawei	Araştırma Mühendisi

Yayın ve Patent Listesi:

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

- **Çelikkaya, Gökhan,** and Gülşen Eryiğit. "A Mobile Assistant for Turkish." TÜRKİYE BİLİŞİM VAKFI BİLGİSAYAR BİLİMLERİ ve MÜHENDİSLİĞİ DERGİSİ 8.8 (2014).
- **Celikkaya, Gokhan,** Dilara Torunoglu, and Gulsen Eryigit. "Named entity recognition on real data: A preliminary investigation for Turkish." Application of Information and Communication Technologies (AICT), 2013 7th International Conference on. IEEE, 2013.