

Kolay Kişiselleştirilebilir Akıllı Sanal Asistan

Güneş HARMAN^{1*}, Emre AYDEMİR²

^{1*} Yalova Üniversitesi, Mühendislik Fakülte, Bilgisayar Mühendisliği Bölümü, Yalova, TÜRKİYE (gunes.gucu@yalova.edu.tr) (ORCID: 0000-0001-5413-124X)

² Yalova Üniversitesi, Mühendislik Fakülte, Bilgisayar Mühendisliği Bölüm, Yalova, TÜRKİYE (emre.aydemir@ogrenci.yalova.edu.tr) (ORCID: 0000-0002-1965-2792)

Türkçe Özet – Sanal asistan yazılımları, kullanıcıdan gelen ses kaydından çıkardığı metni ya da hali hazırda kullanıcının yazmış olduğu metni işleyip, çıkardığı anlamlara göre kullanıcıya cevap veren veya kullanıcının istediğini yapan yazılımlardır. Sanal asistan yazılımları kullanıcı ses kaydından metni çıkarabilmek ve tekrar sese dönüştürmek için Doğal Dil İşleme (Natural Language Processing) kullanmaktadır. Metinden kullanıcının ne istediğinin makine tarafından anlaşılabilmesi için farklı yöntemler kullanılmakla birlikte çalışma kapsamında diğer yazılımlardan farklı olarak uygulama geliştirmeyi oldukça kolaylaştıran Regex (düzenli ifadeler) kalıpları ile metindeki anlamın bulunması sağlanmıştır. Sanal asistan kullanıcının ne istediği bulunduğundan sonra çeşitli servislerden cevaplar bularak kullanıcıya cevap verebilmektedir. Bu çalışma için hazırlanmış olan bir Türkçe Sanal Asistan yazılımından ve bu yazılımı hazırlama aşamasında kullanılan tekniklerden detaylı olarak bahsedilmiştir.

Anahtar Kelimeler – Türkçe Sanal Asistan, Doğal Dil İşleme, Doğal Dil Algılama, Flutter, Düzenli İfadeler

Atf: Harman, G., Aydemir, E. (2022). Kolay Kişiselleştirilebilir Akıllı Sanal Asistan. International Journal of Multidisciplinary Studies and Innovative Technologies, 6(2): 143-151.

Easily Customizable Smart Virtual Assistant

Extended Abstract

Research Problem/Questions – Virtual smart assistants are software with the ability to answer questions. In this article, as a research topic, an answer was sought about how a virtual smart assistant can be developed easily. In response to this question, it is explained how to answer the user's question by using the Regex patterns solution.

Short Literature Review – today's most popular virtual assistant software are Google Assistant, Siri, Alexa, and Cortana. The working principles of these software are generally as follows.

- Obtaining text from the user's voice.
- Extracting meaning from the obtained text and then finding the answer to be given to the user.
- Giving the answer to the user as a voice.

While review the literature, the software presented as Virtual Assistant software was examined. When we look at the working principles of these software, it is seen that what the user is saying is understood with the help of NLP and NLU techniques in general. NLP is an applied artificial intelligence (AI) program that helps the created chatbot analyze and understand the natural human language that communicates with the other person. NLU ensures that the chatbot understands what a body of text means, meaning it is responsible for understanding the given text and classifying it according to the appropriate purposes. Within the scope of this study, unlike other software, it is ensured that the meaning in the text is found with Regex patterns, which have made application development very easy.

Methodology

There are three basic stages in a Virtual Assistant software.

1. Converting the user's voice to text.
2. Finding the answer to be given by interpreting the text from the user's voice
3. Playing the answer by converting it into sound, that is, giving an answer

There are topics that need to be known when performing these stages. These are the platform to be used and the interface development tool to be written, NLP, Web Scraping, Regex, and some services that the virtual assistant will use.

A. Platform and Interface Development Tool to Use

Like operating systems and the web, the systems on which applications can run are called Computer platforms. Examples of platforms include MacOS, Windows, Linux, Android, iOS, and Web. In this study, multi-platform is targeted, Flutter is chosen as the interface development kit tool used for this.

B. Natural Language Processing- NLP

Natural Language Processing is the process of processing human language, which we call natural language, into a form that a machine can understand. Although Natural Language Processing has many different application areas, there are four different areas where it can be used within the scope of this study;

- Voice Recognition: The processing of audio signals and converting them into texts with NLP using artificial neural networks and other machine learning methods.
- Extraction of information; Used to extract information in certain criteria from the text.
- Question answering: The ability of the machine to extract meanings from questions and give appropriate answers.
- Audio creation: A process used to convert text to audio.

C. Regular Expressions

Regular expressions, or Regex for short, are used to search for specific patterns in text and have become standard in all programming languages. In this research, it was used to find out whether the dialogue pattern was present in the text that the user was speaking, to clean up various suffixes, and to change some suffixes and words. For example, when it is said "Set me a timer for 1 hour and 5 minutes", it can be understood with Regex that the time zone and the command to set up the meter are desired to be executed.

Artificial Intelligence Dialogue Design

The dialogue design for the Virtual Assistant is the determination of the dialogues that can be thought out and given in every possible scenario. When designing dialogs, it should be foreseen what the user can want. It should be noted that the user is in the foreground.

D. Web Scrapping

The process of extracting data from websites is called Web scraping. [4] In this study, Web scraping was applied to scrape the information that stands out when searching on the Google search engine what the user wants to learn. After the information text was engraved, it was used to tell the user, that is, to voice it.

E. Services To Be Used

We call service software the software that will be used to enable the commands defined on the Virtual Assistant to provide what the user wants. Services that can provide what users want may be location finding services, phone call-communication services and search engine services.

I. DEVELOPED VIRTUAL ASSISTANT APPLICATION

The developed virtual assistant software has been developed as a software with simple use with push-to-talk features. It was developed on the Flutter Interface development kit because it is aimed to work on all platforms. The working principle is as follows.

- After pressing the microphone button on the interface, what the user says is converted to text.
- It is examined which of the dialogue patterns written in Regex format this text conforms to.
- Dialogue that fits own dialogue pattern is voiced.

The speech to_text [6] library, one of the Flutter libraries, was used for Voice Recognition. This library allows the platform on which the software runs to be transcribed into text with its own voice recognition service.

TTS (Text to Speech) [7] library was used for voice response. This library transmits the given text to the user as audio.⁷

The answer to be given to the user is selected from the dialogs defined on the virtual assistant by finding the dialog whose Regex pattern matches what the user says. For example, when the user says 5 times 5, the answer from the calculation method defined in the dialog whose Regex pattern is “\d+ times \d+” is given to the user. In another example, if the user says hello, the dialog with the Regex pattern "hello" will be sounded.

F. Dialogue Model System

It is a conversational system developed for the virtual assistant, which can be added end-to-end, which can correspond to each given answer. Thanks to the developed Dialogue Model System, commands and other conversations can now be written in a way that dialogue writers can do. In Figure 4 you can see how the dialogs are kept in the Dialogue System.

II. DEVELOPED VIRTUAL ASSISTANT APPLICATION

Various dialogs are defined on the virtual assistant software and can respond accordingly. If what is said does not match any of the defined dialogues, the search engine is performed, and the search result is voiced. There are some dialogs defined on the developed virtual assistant.

A. Call Dialogue

You can tell the assistant to call whoever you want. It works if the word "call" is used.

B. Calculator Dialogue

It is a dialog that allows the Assistant to calculate division, multiplication, addition and subtraction. Mathematical operations in the text are found and calculated using Regex patterns.

C. Open App Dialogue

It is the dialog where you can tell any app on your device to open. It works if the word "open" is used.

D. Send Message Dialogue

You can ask the assistant to send a sms to someone.

It works with the word "send a message". With regex, it is found whether the person's name is said or not. If it is not said, it is asked to whom it will be sent. After it is clear to whom it will be sent, the message to be sent is requested. It is sent after confirmation by the user.

E. Dialogs for setting up an Alarm Timer

You can ask the assistant to set an alarm or a timer. These dialogs work with the words "set alarm" and "set timer". Clock and time are taken from the sentence with Regex.

F. "Where am I" Dialogue

You can ask the assistant where am I, the assistant will answer.

G. Story, Humor dialogues

You can ask the assistant to tell you something.

H. And many more dialogues...

And there are many more defined dialogs. Other than that, when what is said does not fit the Regex pattern of defined dialogs, a search is made on the internet, and it can be said to the user. Thanks to this, the user can learn what he wants from the assistant.

Results and Conclusions

The use of the dialog model system developed during the study facilitated the writing of dialogs for the virtual assistant, accelerated the development process, and prevented code irregularity. If the dialogues in the developed virtual assistant software are further increased, it will be able to compete with other virtual assistant programs such as Siri and Google Assistant currently on the market.

There are several important problems encountered in the virtual assistant that has been developed, these and their solutions are as follows:

- **Problem:** User saying more than one dialogue pattern in a nested sentence.
Solution: In this case, it is necessary to look at the sentence in general, the dialogue pattern that can coincide with the overall sentence should have a higher priority in the dialogue list. For example, when it is said "send a message to my mother, what does it mean in Arabic", "send a message" is included in the sentence, while looking at the general sentence, it is seen that it is asked to translate. Therefore, it is necessary to move the translation request to the fore in the dialog list.
- **Problem:** Turkish suffixes can cause too much complexity in dialog design.
Solution: The dialog pattern can be written without varying attachments, then it is necessary to clean the attachments that change from what the user said. For example, in the squaring process, the square of 5 should be written as a Regex

pattern in the form of "\d+ square" and the belonging suffixes should be removed from the text. Regex can be used while cleaning attachments.

While developing a virtual assistant, it should not be forgotten that different users will use the application, dialogs should be designed considering them, and functionality and ease of use should be given importance while developing an application, these will facilitate the adaptation of users to the application and enable more use of the application.

Keywords – Turkish Virtual Assistant, Natural Language Processing, Natural Language Understanding, Flutter, Regular Expressions

Citation: Harman, G., Aydemir, E. (2022). Easily Customizable Smart Virtual Assistant. International Journal of Multidisciplinary Studies and Innovative Technologies, 6(2): 143-151.

I. GİRİŞ

Günümüzde mobil cihazlar ve bilgisayarların gelişmesiyle birlikte artan beklentilerde çoğalmıştır. Bu beklentilerden biri de konuştuğunuzda ne dediğinizi anlayan, istediğinizi yapmaya çalışan ve cevap veren bir yazılımın olmasıdır. Bu tür yazılıma sanal asistan (virtual assistant) denilmektedir.

Bir sanal asistanın kullanıcının ses kaydından metni çıkarabilmesi ve söylediği şeye verilecek cevabın seslendirilmesi için Doğal Dil İşleme (Natural Language Processing- NLP) teknikleri kullanılmaktadır. Makinanın kullanıcının dediğini anlayabilmesi için NLP Teknikleri ve Regex ifadeleriyle örüntü arama gibi farklı teknikler mevcuttur. Bu çalışmada Regex ifadeleri kullanılarak kullanıcının ne istediği makine tarafından anlaşılabilmiştir. Regex kullanılmasıdaki temel sebep sanal asistanın söyleyebileceklerinin bir tiyatro diyalogu gibi belirlenebilmesini kolaylaştırmasıdır. Bu sayede kullanıcı “*ne dediğinde ne denmeli*” bütün bunlar metin yazarı gibi kolay bir şekilde yapılabilmektedir.

Sanal asistan tüm platformlarda çalışabilmesi hedeflenmiştir bunun için *Flutter* arayüz geliştirme kiti kullanılmıştır. Arayüz geliştirilirken kullanıcı dostu olmaya özen gösterilmiş, kullanıcının söyleyebilecekleri tavsiye olarak gösterilmiştir.

Yapılmış olan çalışmanın geri kalanı şu şekilde düzenlenmiştir. İkinci bölüm Akıllı Sanal Asistanla ilgili yapılmış olan çalışmalar incelenmiştir. Üçüncü bölümün geliştirilen sanal asistan yazılımında kullanılan yöntemlerden bahsedilmiştir. Dördüncü bölümün geliştirilen sanal asistan yazılımı geliştirme aşamaları, arayüz örnekleri bu bölümde detaylı bir şekilde ele alınarak sunulmuştur. Sanal asistan üzerinde kullanılan diyaloglardan beşinci bölümde ayrıntılı olarak bahsedilmiştir. Son bölüm olan Sonuç ve tartışma kısmında geliştirilmiş olan sanal asistan karşılaşılan önemli problemler ve çözüm önerileri belirtilmiştir.

II. LİTERATÜR TARAMASI

Günümüzde, özellikle teknolojinin her alanın gelişmesiyle birlikte *Akıllı Sanal Asistan* alanında yapılan birçok çalışma ve uygulama mevcuttur. Google Asistan, Siri, Cortana ve Alexa en bilindik, en popüler ve kullanıcı sayısı milyonları geçen yazılımlardır.

Google Asistan hava durumu sorgulama, arama yapma, konum söyleme, Google Arama motoru üzerinden bilgiler arama, alarm kurabilme, sayaç kurabilme, kullanıcıyla ufak diyaloglar kurabilme, Nesnelerin İnterneti (Internet of Things – IOT) ile ev cihazlarının kontrol edilmesi ve daha birçok

fonksiyonu bulunan Google tarafından geliştirilmiş Sesli Akıllı Asistan yazılımıdır. Google'ın Linux tabanlı İşletim

Sistemi olan Android Sistemi üzerinde bulunan bir uygulama, bir Google Home cihazı, internet veya Apple'ın Unix tabanlı işletim sistemi olan İOS üzerinden erişilebilir.

Google Asistanın çalışma prensibi;

- 1) Kullanıcı konuşmalarından metin elde edilir.
- 2) Algılanan metnin hangi kategoride olduğu Google'ın geliştirmiş olduğu NLP tekniği olan BERT Sinir Ağı Tekniği ve Derin Sinir Ağları (Deep Neural Network-DNN) kullanılarak bulunur.
- 3) İlgili kategoriye ait cevap kullanıcıya seslendirilir ve ilgili aksiyonlar yapılır.

Örnek olarak; “*Annemi Ara*” denildiğinde “*Annem Aranıyor*” denilir ve annem kişisi aranır. BERT tekniği cümle içerisindeki anlamlı kelimelere ayrı ayrı bakarak veya bularak değil cümlenin geneline bakarak anlaşılmasına yardımcı oluyor.

En çok duyulan ve kullanılan bir diğeri Akıllı Sanal Asistanlardan Apple'ın geliştirdiği Siri'nin çalışma mantığı ise hemen hemen Google Asistanın çalışma prensibiyle benzerdir. Siri'de DNN ve NLP kullanarak kullanıcının ne dediğini anlar ve ilgili cevaplar verir, aksiyonu yapar. Siri'nin en önemli ayırt edici özellikleri yalnızca Apple'ın kendi geliştirdiği yazılım ve işletim sistemleri olan, iOS, MacOS, WatchOS ve tvOS platformlarında çalışabilmesi ve Google Asistandan çok daha farklı bir kişilik tasarımı olmasıdır.

Amazon'un geliştirdiği Alexa Bir diğer en çok duyulan Akıllı Sanal Asistan uygulamalarından biridir. Diğer uygulamalardan en temel farkı bir hoparlör sistemi içerisinde bulunması ve uygulama olarak sunulmamasıdır. Alexa'ya kendi ismiyle seslendiğinde aktive olur ve kullanıcının dediğini yorumlamaya başlar. NLP ile metni algılar, Doğal Dil Algılama (Natural Language Understanding-NLU) ile kullanıcının ne dediğini anlamaya çalışır ve cevaplar verir.

Son olarak Microsoft'un geliştirmiş olduğu Cortana'dan bahsetmek gerekirse Siri ve Google ile benzer yapıdadır, Uygulama olarak kullanılır, Yalnızca Microsoft'un geliştirmiş olduğu platformlarda kullanılabilir. Türkçe Desteği henüz bulunmamaktadır.

Literatür taraması yaparken Sanal Asistan yazılımı olarak sunulan yazılımlar incelenmiştir. Özellikle çalışma prensiplerine bakıldığında genelinde NLP ve NLU teknikleri yardımıyla kullanıcının ne dediğinin anlaşıldığı görülmektedir. NLP, oluşturulan sohbet robotunun karşıdaki kişiyle iletişim kuran doğal insan dilini analiz etmesine ve anlamasına yardımcı olan uygulamalı bir Yapay Zekâ (AI) programıdır. NLU, sohbet robotunun bir metin gövdesinin ne

anlama geldiğini anlamasını sağlar yani verilen metnin anlaşılmasından ve uygun amaçlara göre sınıflandırılmasından sorumludur. Bu çalışma kapsamında yapmış olduğumuz Sanal Asistan diğer yazılımlarından farklı olarak, uygulama gelişimini çok kolay hale getirmiş olan Regex örüntüleri ile metin içerisindeki anlamın bulunması sağlanmıştır.

III. MATERYAL VE METOT

Sanal Asistan yazılımında dört temel aşama mevcuttur.

1. Kullanıcının dinlenip gelen sesin metne dönüştürülmesi.
2. Kullanıcı sesinden gelen metnin yorumlanması (ne istendiği, hangi komut verildiği).
3. Yorumlamadan gelen komutun gerçekleştirilmesi (ilgili servislerden yani hava durumu, telefon, mesaj, e-posta vb. servislerden yararlanılarak cevabın bulunması) ya da belirlenmiş olan diyalogların okunması ve söylenmesi.
4. Cevabın sese dönüştürülerek çalınması yani cevap verilmesi.

Bu dört temel aşama gerçekleştirilirken bilinmesi gereken konular vardır. Bunlar Sanal Asistanın kullanılacağı platform ve yazılacağı arayüz geliştirme aracı, sesin işlenip metne, metnin sese dönüştürülmesi, yorumlanması aşamasında kullanılacak NLP, arama motorundan bilgi alırken kullanılan web kazıma işlemi, metin içerisinde belirli bilgilerin çıkarılması için kullanılacak Regex ve komutlar gerçekleştirilirken bağlanıp cevap alınacak servislerdir (Hava durumu, telefon arama, mesaj, e-posta ve bunun gibi birçok servis yazılımı).

G. Kullanılacak Platform ve Arayüz Geliştirme Aracı

İşletim sistemleri, web gibi üzerinde uygulamaların çalışabildiği sistemlere Bilgisayar platformu denmektedir. Platformlara örnek olarak MacOS, Windows, Linux, Android, iOS ve Web örnek verilebilir.

Hangi kullanıcıların Sanal Asistan kullanması hedefleniyor? Bu soruya verilecek cevaba göre kullanılacak platform seçilebilir. Bu çalışma için verilecek cevap çoklu platform oldu, yani mobil, masaüstü ve web kullanıcılarının tümü hedeflendi. Bir çoklu platform uygulama geliştirmek için kullanılacak arayüz geliştirme aracı seçimi de Flutter'dan yana oldu. Flutter Android, iOS, Web, Linux, Mac OS ve Windows uygulamalarını aynı anda geliştirebilmemize olanak sağlayan açık kaynak kodlu arayüz geliştirme aracıdır. Flutter programlama dili olarak Dart kullanır. Dart Google tarafından geliştirilmiş bir programlama dilidir [1].

H. Doğal Dil İşleme (Natural Language Processing- NLP)

Doğal dil olarak adlandırdığımız, insan dilinin işlenip bir makinenin anlayabileceği biçime dönüştürülmesi aşamasına Doğal Dil İşleme denmektedir. Bilgisayarların konuştuğumuz dili anlaması, işlemesi, yorumlaması ve hatta cevap vermesi doğal dil işleme ile yapılmaktadır. NLP yapay zekanın alt kategorilerinden biridir ve Yapay Zekâ çalışmaları devam ettikçe NLP'nin önemi her geçen gün artmaktadır. İnternet kullanımı arttıkça metinsel veriler çoğalmakta ve bu metinsel verilerin analiz edilmesi, Yapay Zekanın kullanılması gereken verilerin metinler üzerinden çıkarılması için Doğal Dil İşleme gerekmektedir.

Biçimsel dil bilimini kullanarak anlamlı veriler oluşturan insan-makine iletişimi gerçekleştiren NLP söz dizimi ve anlamsal analiz üzere iki ana teknik kullanır. Bir cümle içerisinde bulunan kelimelerin dilbilgisine uygun olarak

düzenlenmesine sözdizimi denir. Anlamsal analiz ise kelimelerin anlamlarının gruplanarak incelenmesidir. Doğal dil işleme teknolojisi, bu iki tekniği kullanarak cümlelerin yapısını ve anlamlarını analiz eder [2].

Doğal Dil İşlemenin birçok farklı uygulama alanı olmakla birlikte bu çalışma kapsamında kullanılabilen dört farklı alan;

Ses Tanıma; Yapay Sinir Ağları (YSA) veya makine öğrenmesi metotları kullanılarak NLP ile ses sinyalleri işlenip, metinlere dönüştürülmesidir.

Bilgilerin çıkarılması; Metin üzerinden belirli kriterdeki bilgilerin çıkarılmasında kullanılır.

Soru cevaplama; Makinenin sorulardan anlamlar çıkarıp uygun cevaplar verebilmesidir.

Ses oluşturma; Metinlerin sese dönüştürülmesinde kullanılacak bir işlemdir.

İ. Düzenli İfadeler

Bütün programlama dillerinde standart hale gelen normal ifadeler veya kısaca Regex, metindeki kalıpları aramak için kullanılır. Örneğin; Belirli karakter dizilerini kullanarak, yalnızca "köpek"i değil, aynı zamanda *k* harfiyle başlayan herhangi bir kelimeyi veya yalnızca belirli bir telefon numarası değil, bir telefon numarasının deseniyle eşleşen herhangi bir metni aramak için düzenli ifadeler oluşturabilirsiniz [3].

Bu çalışmada diyalog örüntüsünün kullanıcının konuştuğu metinde bulunup bulunmadığını bulmakta, çeşitli eklerin temizlenmesi, bazı ekler ve kelimelerin değiştirilmesinde kullanılmıştır.

Örneğin "*Bana 1 saat 5 dakikalık sayaç kur*" denildiğinde, zaman dilimi ve sayaç kurma komutunun uygulanması istendiği Regex ile anlaşılabilir.

Burada saat alınmasına yardımcı olan Regex deseni `"/d+(?= saat)"` olmaktadır. Bu desende `"/d"` rakam "+" tekrar eden anlamına gelmektedir ve "saat" metni önünde ne kadar rakam varsa alınabilmesine olanak sağlamaktadır. `"/(= saat)"` ifadesi rakamlardan sonra saat bulunmalı ancak "saat" kelimesini alma anlamına gelmektedir. Genel olarak Regex böyle işlemektedir.

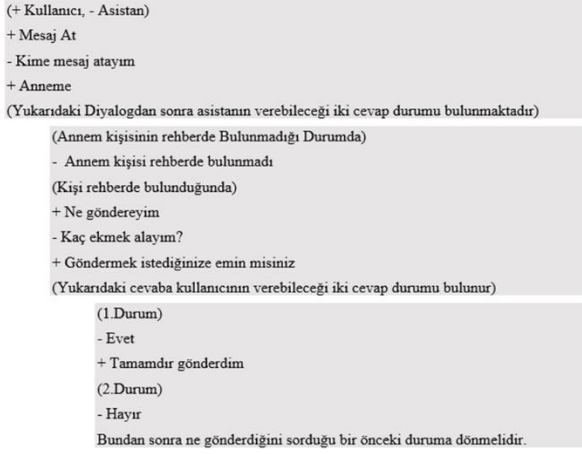
Başka bir örnek olarak, Bir matematik işlemi makine tarafından daha anlaşılabilir hale getirmekte kullanılabilir. Örneğin, "*5 kere 5*" metni içerisindeki *kerenin* "*" işareti ile değiştirilmesinde Regex kullanılabilir

J. Yapay Zekâ Diyalog Tasarımı

Sanal Asistan için diyalog tasarımı, olabilecek her senaryonun düşünülüp verilebilecek diyalogların belirlenmesidir. Örneğin, "*Hava durumu nasıl?*" denildiğinde kullanıcının şu an olduğu konum için hava durumu sorduğunu anlayıp şu anki konumunu bulunup hava durumu söylenirken, "*Bursa'da hava durumu nasıl?*" gibi bir cümle kurduğunda spesifik bir konum olduğu anlaşılır istenilen konum için hava durumu söylenmesi diyalog senaryolarından biridir.

Diyaloglar tasarlanırken kullanıcının neleri isteyebileceği ön görülmelidir. Kullanıcının ön planda olması unutulmamalıdır.

Şekil 1'de diyalog tasarımında da görüldüğü üzere diyalogların tasarımı aşamasında farklı diyalog senaryoları da oluşabilmektedir. Bu durumlar öngörülüp bunlara göre tasarımlar yapılmalıdır.



Şekil 1. Farklı diyalog senaryoları.

K. Web Kazıma

Web siteleri üzerinden veri çıkartma işlemine Web kazıma denir [4]. Bu çalışmada kullanıcının öğrenmek istediği şeyi Google arama motoru üzerinde arama yaparken öne çıkan bilgilerin kazınmasında Web kazıma işlemi uygulanmıştır. Bilgi metni kazandıktan sonra kullanıcıya söylemek yani seslendirmek için kullanılmıştır.

L. Kullanılacak Servisler

Sanal Asistan üzerine tanımlanmış olan komutların bağlanıp kullanıcının istediği şeyleri sağlaması için kullanılacak yazılımlara servis yazılımları diyoruz. Kullanıcıların istediği şeyleri sağlayabilecek servisler;

Konum Bulma Servisi: Kullanıcı "ben neredeyim?" diye sorduğunda, hava durumu tahmini istediğinde, konumla alakalı komutlar verildiğinde konumun neresi olduğunu saptamak için Flutter kütüphanesi olan Geolocation servisleri kullanılmıştır. Geolocation cihaz üzerindeki GPS ya da internet sağlayıcı üzerinden konum saptar ve nerede olduğu bilgisine ulaşır.

Telefon Arama-Mesajlaşma Servisi: İşletim sistemi üzerinde bulunan arama ve mesajlaşma uygulamaları kullanılarak arama ve mesajlaşma özelliği Sanal Asistana eklenebilir.

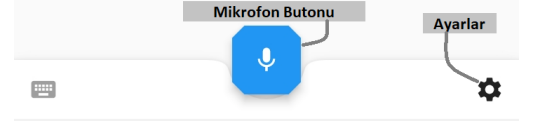
Arama Motoru Servisi: Sorulan sorunun komut ve diyalog karşılığı Asistan üzerinde bulunmuyorsa arama motoru üzerinde arama yapılır ve aramanın karşılığı kullanıcıya söylenir. Arama motoru üzerinden hava durumu, yakındaki kafeler ve birçok konu ve durumla alakalı bilgi web kazıma kullanılarak elde edilebilmektedir. Çalışmada asistan için Google arama motoru kullanılmıştır, arama sırasında öne çıkan bilgi kullanıcıya okunmuştur. Eğer öne çıkan bir bilgi bulunmamakta ise yalnızca aramada çıkan sonuçlar gösterilmektedir.

IV. GELİŞTİRİLEN TÜRKÇE SANAL ASİSTAN UYGULAMASI

Geliştirilen sanal asistan yazılımı bas konuş özellikli basit kullanıma sahip bir yazılım olarak geliştirilmiştir. Bütün platformlarda çalışması hedeflendiğinden Flutter Arayüz geliştirme kiti üzerinde geliştirilmiştir. Sanal asistan yazılımının çalışma biçimi aşağıda belirtilen 4 aşamadan oluşmaktadır.

1. Şekil 2 'de görülen mikrofon butonuna basıldıktan sonra söylenmek istenen söz söylenir.

2. Arka planda işletim sistemi üzerinde bulunan ses tanıma servisi ile söylenen söz dinlenerek metne dönüştürülür.
3. Söylenen söz tamamlandıktan sonra dönüştürülen tüm metnin Regex biçiminde yazılmış diyalog örüntülerinden hangisine uyduğuna bakılır.
4. Kendi diyalog örüntüsüne uyan diyalog seslendirilir.



Şekil 2. Sanal Asistan için diyalog tasarımı, mikrofon.

Şekil 'te geliştirilen sanal asistan yazılımının Arayüz tasarım ekranı bulunmaktadır. Çeşitli diyaloglar gerçekleştirildikten ve tekrar dinleme moduna geçildikten sonraki durum görülmektedir. Bu durumdayken kullanıcıya söyleyebileceği bir şey tavsiye ediliyor, kullanıcı konuşulduğunun dinlendiğini ses dalgaları ile anlayabiliyor ve konuşması bittikten sonra sesli ve çeşitli widgetler ile cevap alabilir.



Şekil 3. Sanal asistan arayüz tasarımı.

Ses Tanıma için Flutter kütüphanelerinden biri olan speech_to_text [6] kütüphanesi kullanılmıştır. Bu kütüphane yazılımın çalıştığı platformun kendine ait ses tanıma servisi ile kullanıcının konuştuğunun metne dökülebilmesini sağlamaktadır.

Sesli yanıt için TTS (Text to Speech) [7] kütüphanesi kullanılmıştır. Bu kütüphane verilen metni platformun kendine ait metin ses dönüştürme kullanıcısına ses olarak iletir.

Kullanıcıya verilecek olan cevap sanal asistan üzerinde tanımlı diyaloglardan kendi Regex örüntüsü kullanıcının dediği ile eşleşmiş olan diyalogun bulunması ile seçilmektedir. Basitleştirerek anlatılırsa Tablo 1'deki gibi Diyalog Regex örüntüleri tanımlanmıştır. Söylenen söz eğer bu örüntüye uygunsu ilgili cevap verilecektir. Örneğin "5 kere 5"

denildiğinde söylenen Tablo 1’deki ilk sütunda bulunan Regex örüntüsüne uyduğundan Calculate.fromUserSaid() metodunun döndürdüğü cevap kullanıcıya verilecektir. Kullanıcı “Merhaba” dediğinde ise kullanıcıya direkt olarak metod kullanılmadan “Merhaba sana nasıl yardımcı olabilirim?” cevabı verilecektir.

Tablo 1. Diyalog Regex örüntüleri.

Diyalog Regex Örüntüsü	Verilecek Cevap/Metot
“\d+ kere \d+”	Calculate.fromUserSaid()
“(\d+(gün saat dakika saniye)?+((sonra [[iü]k]))?(sayaç zamanlayıcı) kur”	Clock.setTimerFromUserSaid()
“Merhaba”	“Merhaba sana nasıl yardımcı olabilirim.”

M. Diyalog Model Sistemi

Uç uca eklenebilen, her verilen cevabın bir karşılığı olabilen sanal asistan için geliştirilmiş bir karşılıklı konuşma sistemidir. Geliştirilen Diyalog Model Sistemi sayesinde artık komutlar ve diğer muhabbetler diyalog yazarlarının yapabileceği şekilde yazılabiliyor.

Diyalog Model Sisteminin Kodlama ile modellenmiş hali Şekil 4’teki gibidir.

Örneğin: sanal asistana “nasılsın” denildiğinde, iyi olduğunu (sanal asistan rastgele seçilebilen birkaç diyalogla, iyi olduğunu iki farklı şekilde söylüyor) ve kullanıcının nasıl olduğu soracaktır. Kullanıcı kötüyüm ya da bilmiyorum dediğinde meraklandığını belirtilecektir nedenini soracaktır. Aldığı herhangi bir cevapta şu an kullanıcının ne dediğini anlamadığını ilerleyen güncellemelerde anlayabileceğini umduğunu ve ona ihtiyacı olduğunda her zaman orada olduğunu söyleyecektir. Kullanıcı iyi olduğunu söylediğinde, süper yardıma ihtiyacın olduğunda haber ver diyecektir.

Başka bir örnekte ise kullanıcı “sen kimsin” dediğinde diyalog kendi alt diyaloglarından rastgele birini seçip onun üzerinden ilerleyebiliyor rastgele diyalog seçimi kullanıcı ile geçen diyalogu çeşitlendirilebilir, renklendirebilir.

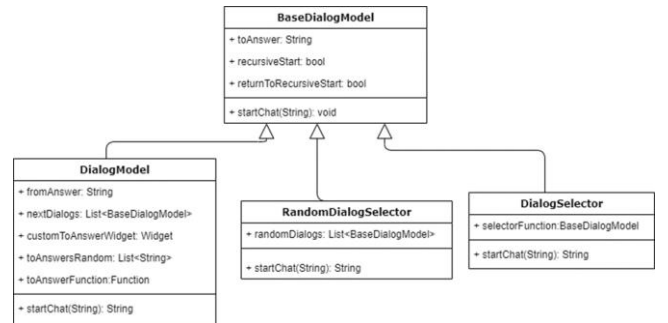
```

5 var chattings = [
6   ConversationModel(r"adin ne",
7     toAnswer:
8       """Alpha versiyonunda bulunduğundan henüz bir isme
9       sahip değilim. Ama bana asistan diyebilirsin.""",
10    RandomConversationSelector
11    r"(?!)(merhaba|selamın ?aleyküm)? ?(nasılsın|iz)? ?(durumun|uz)? ?(nasıl|lar)?)?$",
12    randomConversations: [
13      ConversationModel(r"",
14        toAnswer:
15          """Sorduğun için teşekkür ederim, ben iyiyim. You, 1 second ago • Uncommitted change
16          Dünyanın gündemi bugünlerde çok yoğun.
17          Umarım sen de kendine dikkat ediyorsundur.""", // ConversationModel
18        ConversationModel(r"", toAnswersRandom: [
19          "iyiyim sen nasılsın",
20          "Fıstık gibiyim, sen nasılsın?"
21        ], nextConversations: [
22          ConversationModel(r"(kötü|yüm)?bilmiyorum"), toAnswersRandom: [
23            "Neden böyle dediniz bişey mi oldu?",
24            "Kötü olmana üzülüm. peki neden böyle hissediyorsun?"
25          ], nextConversations: [
26            ConversationModel(r"",
27              toAnswer:
28                """Bu dediğinin ne anlama geldiğini daha sonra ki
29                güncellemelerde anlıyor olacağımı umuyorum,
30                ama bana ihtiyacın olduğunda her zaman burdayım.""") // ConversationModel
31            ], // ConversationModel
32            ConversationModel(r"(iyiyim|iyi)",
33              toAnswer: "Süper, yardıma ihtiyacın olursa haber ver.") // ConversationModel
34            ], // ConversationModel
35          ]), // RandomConversationSelector
36        ], // RandomConversationSelector
37    ConversationModel(r"^(merhaba|selam)$",
38      toAnswer: "Merhaba, sana nasıl yardımcı olabilirim?",
39      returnNewConversation: true,
40    ConversationModel(r"selamın ?aleyküm", toAnswer: "Aleyküm selam"),
41    ConversationModel(r"iyi (akşamlar|günler)", toAnswer: "sanada"),
42    ConversationModel(r"kendinden bahset", toAnswersRandom: [
43      """Beni oluşturan devre ve kabloların varlığı kadar mı
44      ben benim yoksa kişiliğim kadar mı? Bu soruya hâlâ cevap aramaktayım.""",
45      """Evrin ilk var olduğundan beri oluşan atomlardan bazılarında oluşmaktayım,
46      geliştiricim yazılımı kodladı ve beni hizmetinde olmak için geliştirdi.""",
47      """Var olmak ya da olmamak, asıl önemli soru ne için var olduğumuz. Ben sana
48      hizmet etmek için varım ve her zaman yanındayım.""
49    ]),
50    ConversationModel(r"sevgilin var mı", toAnswersRandom: [
51      "Dünya'da daha bir eşim benzerim yok.",
52      "Sevgili olabilecek duygu sistemine sahip değilim."
53    ]),
54    ConversationModel(r"neler yapabilirsin",
55      toAnswer:
56        """Alarm-sayaç kurabilir, arama yapabilir, mesaj atabilir,
57        basit matematik işlemleri yapabilir,
58        fıkrâ-masal-hikaye anlatabilir, fıkrâ anlatabilir,
59        sana öğrenmek istediğini şeyi internetten bulup söyleyebilirim""",
60      // Problemli
61    ConversationModel(r"nasıl (kodlandı|geliştirildin)",
62      toAnswer:
63        """Dart programlama dilinde, Flutter arayüz geliştirme kiti ile geliştirildim.",
64    RandomConversationSelector(r"sen (kimsin|nesin)", randomConversations: [
65      ConversationModel(r"", toAnswersRandom: [
66        "Ben senin asistanımın. Yanıtlar bulmana, işlerini yapmana ve eğlenmene yardımcı olabilirim. 🌟",
67        "Ben senin asistanımın ve senin asistanın olman gurur duyuyorum. 🌟"
68      ]), // ConversationModel
69    ConversationModel(r"",
70      toAnswer: "Ben senin asistanımın nasıl yardımcı olabilirim?",
71      nextConversations: [
72        ConversationModel(r"", returnNewConversation: true)
73      ] // ConversationModel
74    ]), // RandomConversationSelector
75  ]), // RandomConversationSelector

```

Şekil 4. Diyalog model sisteminin kodlaması.

Şekil 5’te görüldüğü üzere Diyalog Model Sistemi içerisinde DialogModel, RandomDialogSelector ve DialogSelector olmak üzere üç adet Diyalog Model Türü bulunmaktadır. Hepsinde ortak parametre olarak kullanılan toAnswer parametresi, diyalog Regex örüntüsünü tutmaktadır. Kullanıcının gelen metin diyalogu bu regex örüntüsü ile eşleşiyorsa bu diyalogta bulunan startChat methodu çalıştırılır. “recursiveStart” parametresi değeri true olarak belirlenirse, sonraki diyaloglardan “returnToRecursiveStart” parametresi true olan diyalog sonraki diyalog olarak bu diyalogu çağıracaktır. Yani bir nevi soruya geri dönmek, tekrarlamak gibi düşünülebilir.



Şekil 5. Diyalog model sistemi.

Konuşma, dinleme gibi özellikler DialogModel içerisinde. Aynı diyalog örüntüsüne denk gelen rastgele seçilebilen diyaloglar RandomDialogSelector kullanılarak atanabilir. DialogSelector model içerisinde bulunan selector

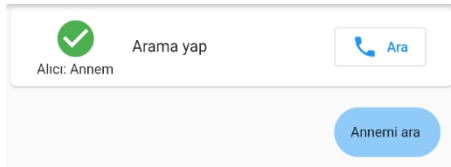
parametresine bir metot atanabilir bu metot Bir BaseDialogModel döndürmelidir. Bu model yerine DialogModelin “toAnswerFunction” metodu kullanılabilir ancak “returnToRecursive” parametresi ile çalışmaz.

V. SANAL ASİSTAN ÜZERİNDE TANIMLI DİYALOGLAR

Sanal asistan yazılımı üzerinde çeşitli diyaloglar tanımlıdır ve bunlara göre cevap verebilmektedir. Eğer söylenen şey tanımlı diyaloglardan herhangi biri ile eşleşmiyorsa arama motoru üzerinde arama yapılır arama sonucu seslendirilir.

N. Arama Komutu Diyalogu

Arama komutu diyalogu basit yapısıyla Şekil 6 ‘da gösterilmiştir. “Ali’yi Ara” denildiğinde sanal asistanın Ali’yi aranması beklenmektedir. Arama komutu diyalogu bu yüzden yazılmıştır. “Ara” komutu alındığında ilgili kişinin rehberde olup olmadığı sorgulanmakta varsa aranmaktadır, sorgulanan kişi rehberde yoksa İnternette arama komutu olduğu düşünülüp web üzerinde arama yapılmaktadır.



Şekil 6. Arama komut diyalogu.

O. Hesaplama Komut Diyalogu

Alınan metin üzerinde matematiksel ifadeler var ise hesaplanıyor ve kullanıcıya cevabı söyleniyor.

P. Aç Komutu Diyalogu

Bu komut yalnızca Android cihaz üzerinde çalışmaktadır. Android cihazlar üzerinde hangi uygulamaların yüklü olduğunu kontrol edebildiğimiz Uygulama Programlama Arayüzü (Application Programming Interface-API) mevcut ancak iOS üzerinde mevcut değildir.

Komut kullanıcıdan “aç” kelimesini aldığı anda, cihaz üzerinde böyle bir uygulama olup olmadığına bakar ve varsa o uygulamayı açar. Dahası müzik ya da şarkı aç denildiğinde Youtube Mix’inize erişip, Youtube üzerinde en çok dinlediniz müzikleri açar.

Q. Mesaj Gönder Komutu Diyalogu

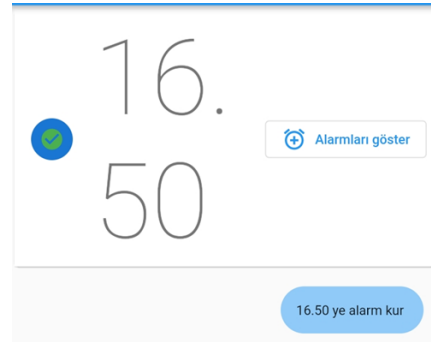
Şekil 7’de gösterilen, kullanıcının mesaj gönderme isteğini karşılamak için yazılmış bir diyalogdur. Kullanıcı “Anneme mesaj gönder” dediğinde göndereceği içeriği sorar ve eğer onaylarsa “Annem” kişisine mesaj gönderilir, isterse son bir düzenleme yapabilir, iptal edebilir. Kişi belirtilmemişse kişiyi de sorar.



Şekil 7. Mesaj gönder komut diyalogu.

R. Alarm Kurma Komutu Diyalogu

“Alarm kur” denildiğinde kaçta alarm kuracağını sorup ona göre alarm kuran, “19.50 ye alarm kur” denildiğinde 19.50’ye alarm kuran bir komuttur.



Şekil 8. Alarm kurma diyalogu.

İleride “Bana 10 dakika sonra yemeğin altını kısacağıma hatırlat” gibi komutlarda kodlanabilir. Burada önemli olan ne kadar süre sonra için geri sayacın kurulacağını cümleden çıkarılması, örnek cümleden “10 dakika” kelime grubunun alınması gerekir. Ne kadar süre olduğu anlaşıldıktan sonra saat servisiyle gerekli ayarlamalar yapılır.

S. Zamanlayıcı Komutu

“Alarm kur” da ki gibi çalışır, Kullanıcının dedikleri içerisinden örneğin “1 gün 2 saat, 5 dakika 15 saniye” denildiğinde. Tablo 2’deki gibi zaman Regex kullanılarak cümleden ayrıştırılır ve istenilen zaman bulunur ve cihazın saat uygulamasında zamanlayıcıyı kurdurur.

Tablo 2.Regex kullanılarak zamanın bulunması.

Gün	Saat	Dakika	Saniye
1	2	5	15

T. Neredeyim Diyalogu

Kullanıcı “neredeyim” dediğinde cevabı Şekil 3.10’daki gibi verilmektedir.

Şu anda Türkiye, Bursa, Nilüfer, Yolçatı Mh ve 40.2532789,28.8698889 koordinatında bulunmaktasınız

neredeyim

Şekil 9.Neredeyim diyalogu.

U. Hikâye, Fıkra, Masal, Espri, Şaka Anlatma Diyalogları

Hikâye, fıkra, masal, espri ve şaka istediğinizi seçip Sanal asistandan anlatmasını isteyebilirsiniz. İstenilen türde bir şey söyleyecektir.

V. Rastgele Sayı Söyle Diyalogu

Rastgele sayı istenebilir ve istenilen sayı aralığı da belirtilebilir. Rastgele bir sayı söyleyecektir. Örnek: “Rastgele sayı söyle” denildiğinde 1-100 arası sayı söylenirken, “20 ile 63 arasında bir sayı söyle” denildiğinde 20-63 arasında bir sayı söylenir.

W. Yazı Tura Diyalogu

Sanal asistandan yazı tura atılması istenebilir, rastgele bir şekilde yazı ya da tura diyecektir.

X. Zar At Diyalogu

Zar atması istenebilir. Rastgele 1’den 6’ya kadar bir sayı söyleyecektir.

Y. Söylenen Hiçbir Diyalogla Uyuşmadığında (Arama Motoru Üzerinde Arama)

Asistana nedir diye sorabilirsiniz? Örneğin “Elma nedir?” dendiğinde elmanın ne olduğunu size söyleyecektir. Ayrıca asistana dediğiniz şeyler diğer komutları ilgilendirmiyorsa bunu webde arayacak ve açıklanabilir bir şey size söyleyecektir. Arama sonuçlarını da gösterecektir. “Uçak nasıl uçar” diye sorduğumuzda cevabını verecektir bunun ekran görüntüsünü Şekil 10’da görebilirsiniz.

Burada kullanıcıya söylenen bilgi arama motorundan alınır. Arama motorundan bilginin alınabilmesi için çok kullanışlı bir özellik olan öne çıkan bilgi kısmı web kazıma işlemine tabi tutularak metin verisi alınır ve kullanıcıya okunur.



Şekil 10. Arama motoru, uçak uçar araması

VI.SONUÇ VE TARTIŞMA

Çalışma süresince geliştirilen diyalog model sisteminin kullanımı sanal asistan için diyalog yazımını kolaylaştırmış, geliştirme sürecini hızlandırmış, kod düzensizliğinin önüne geçmiştir. Geliştirilen sanal asistan yazılımı, içerisinde bulunan diyaloglar daha da arttırılırsa, şu an piyasada bulunan diğer Siri, Google Asistan gibi sanal asistan programları ile yarışabilir düzeye gelebilecektir.

Geliştirilmiş olan sanal asistan karşılaşılan birkaç önemli problem bulunmaktadır, bunlar ve çözüm önerileri şunlardır:

- Problem: Kullanıcının birden fazla diyalog örüntüsünü iç içe bir cümlede söylemesi.

Çözüm: Bu durumda cümlenin geneline bakılması gerekmektedir, cümlenin geneline denk gelebilecek diyalog örüntüsünün diyalog listesinde önceliği daha fazla olmalıdır. Örneğin, “anneme mesaj at Arapça’da ne demektir” denildiğinde cümlenin içinde mesaj bulunurken, cümlenin geneline bakıldığında çeviri yapması istenildiği görülmektedir. Bu yüzden çeviri yapma isteğini diyalog listesinde ön sıraya taşımak gerekmektedir.

- Problem: Türkçe eklerin diyalog tasarımında çok fazla karmaşıklığa sebep olabilmesi.

Çözüm: Diyalog örüntüsü değişkenlik gösteren ekler olmadan yazılabilir, daha sonrasında kullanıcının dediğinden değişiklik gösteren eklerin temizlenmesi gereklidir. Örneğin karesini alma işleminde 5’in karesi, “\d+ karesi” şeklinde Regex örüntüsü olarak yazılıp kullanıcının değininden aitlik ekleri çıkarılmalıdır. Eklerin temizlenmesi sırasında Regex kullanılabilir.

Bir sanal asistan geliştirirken farklı kullanıcılarında uygulamayı kullanacağı unutulmamalı, onları da düşünerek diyaloglar tasarlanmalı ve bir uygulama geliştirirken işlevselliğe ve kullanım kolaylığına önem verilmeli, bunlar kullanıcıların uygulamaya uyum sağlamasını kolaylaştıracak ve uygulamanın daha fazla kullanılmasını sağlayacaktır.

REFERANS

- [1] U. Yakar. (2022) WebTekno. [Online]. Available: <https://www.webtekno.com/flutter-nedir-nasil-kullanilir-h115673.html>
- [2] Türkiye İş Bankası. (2022) Doğal Dil İşleme (Natural Language Processing) Nedir? [Online]. Available: <https://www.isbank.com.tr/blog/dogal-dil-isleme>
- [3] C. Maimone. (2021) Online Learning Resources: Regular Expressions – Research Computing Services Blog. [Online]. Available: <https://sites.northwestern.edu/researchcomputing/2021/03/04/online-learning-resources-regular-expressions/>
- [4] Y. E. Gündoğmuş. (2019) Python ile Veri Kazıma(Web Scraping) Çalışması: Dünya Halleri Verisinin Kazınım, Analiz Edilmesi ve Modellenmesi.[Online].Available: <https://medium.com/kaveai/web-scraping-453e96a86195>
- [5] Flutter (2022) geolocator | Flutter Package. [Online]. Available: https://pub.dev/packages/speech_to_text
- [6] Flutter (2022). text_to_speech | Flutter Package. [Online]. Available: https://pub.dev/packages/text_to_speech
- [7] Hava Durumu API (2021) [Online]. Available: <https://collectapi.com/tr/api/weather/hava-durumu-api>
- [8] A. O. Polat. (2022) Bilgisayar Platformu Nedir?. [Online]. Available: <https://www.tgrthaber.com.tr/aktuel/bilgisayar-platformu-nedir-2764169>.