Umjetni inteligentni agent kao posrednik između čovjeka i izvora podataka

Filinić, Goran

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:709789

Rights / Prava: Attribution 3.0 Unported/Imenovanje 3.0

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Goran Filinić

UMJETNI INTELIGENTNI AGENT KAO POSREDNIK IZMEĐU ČOVJEKA I IZVORA PODATAKA

ZAVRŠNI RAD

Varaždin, 2021.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Goran Filinić

Matični broj: 0016124457

Studij: Informacijski sustavi

UMJETNI INTELIGENTNI AGENT KAO POSREDNIK IZMEĐU ČOVJEKA I IZVORA PODATAKA

ZAVRŠNI RAD

Mentor:

dr. sc. Bogdan Okreša Đurić

Varaždin, rujan 2021.

Goran Filinić

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI Radovi

Sažetak

U završnom radu govorit će se o umjetnom inteligentnom agentu kao posredniku između čovjeka i izvora podataka. Rad se fokusira na *chatbota* koji je kreiran u telegramu s *backendom* u Pythonu. Funkcionalnost *chatbota* je dohvaćanje podataka s API-a o trenutnom stanju COVID-19 virusa. Prilikom korisnikovog unosa, *chatbot* vraća relevantne podatke. Korisnik će imati nekoliko komandnih linija koje pozivaju različite funkcionalnosti s *botom*. Objasniti će se potreba za takvim *botovima* u modernom svijetu te kako poduzeća mogu profitirati od jednog takvog *bota*.

Ključne riječi: umjetna inteligencija, chatbot, telegram, izvori podataka, inteligentni agent

Sadržaj

1. Uvod 1

2. Umjetna inteligencija 2

2.1. (Inteligenti) Agent 2

2.2. Povijest umjetne inteligencije i Turingov test 4

3. Agent za razgovor 7

3.1. Agenti za razgovor kroz povijest 7

3.1.1. Eliza 7

3.1.2. Parry 8

3.1.3. Alice 8

3.1.4. Inteligentni osobni asistenti 8

3.1.5. Agenti za razgovor na društvenim mrežama 10

3.2. Tehnologija 10

3.2.1. Domena znanja 10

3.2.2. Pristupi 11

4. Praktičan dio - integracija *chatbota* u aplikaciju Telegram 15

4.1. Telegram 15

4.2. python-telegram-bot 17

4.3. Implementacija 17

4.4. Primjer korištenja 20

5. Zaključak 24

Popis literature 26

Popis slika 27

Popis tablica 28

Popis isječaka koda 29

1. Uvod

U modernom digitalnom svijetu, pogotovo nakon globalne pandemije SARS-CoV-2 virusa, online trgovina i poslovanje doživjeli su ubrzani razvoj. Tako online poduzeća, kako bi ostala relevantna, moraju sebe istaknuti u svom području i postaviti se kao voditelji svoje branše. Tu u priču dolaze umjetni inteligentni agenti za razgovor , pošto jednostavnim automatizmom mogu korisniku dati korisne informacije, a prilikom toga korisnik konzumira određeni sadržaj poduzeća. Za umjetne inteligente agente za razgovor se kroz cijeli rad koristi engleski izraz *chatbot*. U marketingu, pogotovo kad pričamo o online području zbog količine izbora, što više netko konzumira određeni sadržaj, znači da prati aktivnosti nekog poduzeća te će biti više zainteresiran u kupnji proizvoda ili usluga što nudi to poduzeće. Najjednostavniji primjer *chatbota* za poduzeće bio bi odgovor na neke FAQ *(Freqently Asked Questions)*, u prijevodu često postavljena pitanja, poput: radnog vremena, dostupnosti nekog proizvoda i slično. *Chatbot* može za poduzeća biti i odličan dodatak obliku marketinga posvećenom online sadržaju i promociji marketinga sadržaja.

U sklopu praktičnog dijela izrađuje se *chatbot* na Telegramu pomoću dodatka za python te se dodaje nekoliko funkcionalnosti osim osnovog izmjenjivanja poruka poput ispisa podataka o COVID-19 pandemiji. U teroretskom dijelu objašnjavaju se osnove kako *chatbot* i cijeli programski proizvod funkcionira, ali i kako se marketing sadržaja (eng. content marketing) i *chatbot* mogu koristiti kako bi se dobio određeni angažman korisnika.

Za izradu završnog rada korištene su knjige, internetske stranice, stručni članci i Github sadržaji vezani za temu rada. Pri istraživanju i pronalasku literature naglasak je bio na stručnim radovima vezanim za temu umjetne inteligencije. U praktičnom dijelu završnog rada, odnosno za izradu *chatbota* te integraciju u aplikaciju Telegram korišten je programski jezik Python i dodatne biblioteke preuzete s Githuba.

2. Umjetna inteligencija

Kako bi se u potpunosti razumio *chatbot* i njegov potencijal, moramo prvo istražiti pojam umjetne inteligencije pošto se *chatbot* smatra (više ili manje, ovisno o *chatbot*u i osobi koja ga promatra) umjetnom inteligencijom. "Umjetna inteligencija *eng. Artificial Intelligence (AI)* se sve više integrira u svakodnevni ljudski život kroz stvaranje i analizu inteligentnog *softwarea* te sklopovlja, koje nazivamo inteligenti agenti *eng. intelligent agents*" [1, str. 373].

Umjetna inteligencija koja se danas poznaje može se kategorizirati prema radnim sposobnostima i kompetencijama u sljedeće dvije vrste [2]:

- Slaba umjetna inteligencija; Vrsta umjetne inteligencije s dizajnom za osobnog asistenta, odnosom s korisnicima, video igrama i upitnicima poznatim kao slaba umjetna inteligencija. Sastoji se od algoritma i izvora podataka. Najčešće su spomenuti algoritam i izvor podataka povezani s uslužnom djelatnosti poduzeća koja imaju dovoljnu količinu podataka na temelju kojih se stvaraju algoritmi. Primjeri chatbotova koji se mogu svrstati pod slabu inteligenciju su Amazon Alexa, Appleova Siri ili Samsungova Bixby .
- Jaka umjetna inteligencija; Vrsta umjetne inteligencije koja se može objasniti kao sustav koji obavlja zadatke koje izravno izvode ljudi, poput vožnje automobila. Ova vrsta zadataka je složenija i razmatra se u kompliciranom sustavu. Jasno je da je riječ o kompliciranom sustavu jer vožnja automobila zahtjeva niz sposobnosti te je potrebno poštovati niz pravila koja su uvjetovana situacijama na cesti. To je zapravo program za rješavanje situacija u kojima odluka može značiti život ili smrt, zbog tih se okolnosti jako pažljivo kalibriraju te se zapravo teško testiraju. Ova kategorija umjetne inteligencije može zamijeniti ručni operativni zadatak čovjeka programiranim strojem. Uspješna implementacija ove umjetne inteligencije ukoliko je dobra izvedena ima potencijal pridnoijeti više čovjeku od slabe umjetne inteligencije.

S obzirom na to da je glavno polazište ovog završnog rada umjetni inteligentni agent koji služi kao posrednik između čovjeka i izvora podataka, njega se može navesti kao primjer slabe umjetne inteligencije. Naime, *chatbot* o kojem se govori u završnom radu je primjer *chatbota* baziranog na pravilima, koji korisnik koristi u sklopu određene aplikacije, u slučaju ovog završnog rada: Telegrama.

2.1. (Inteligenti) Agent

"Agent je nešto što djeluje u svojoj okolini - svojim radnjama mora moći utjecati na svoju okolinu". [3, str 4.]. To uključuje svakojake pojmove, poput mačke, insekta, termometra, automobila, robota, čovjeka, organizacije i poduzeća, države, osobnog računala i sl. Svaki od navedenih agenta na neki način djeluje na svoju okolinu, neki naravno manje poput mačke na emocionalno stanje svojeg vlasnika. U svakom slučaju bit proučavanja agenta je što on radi,

odnosno kako se ponaša u svojoj okolini. Prema Pooleu sudimo agentu po njegovim radnjama te se može smatrati inteligentnim ukoliko [3, str 4.]:

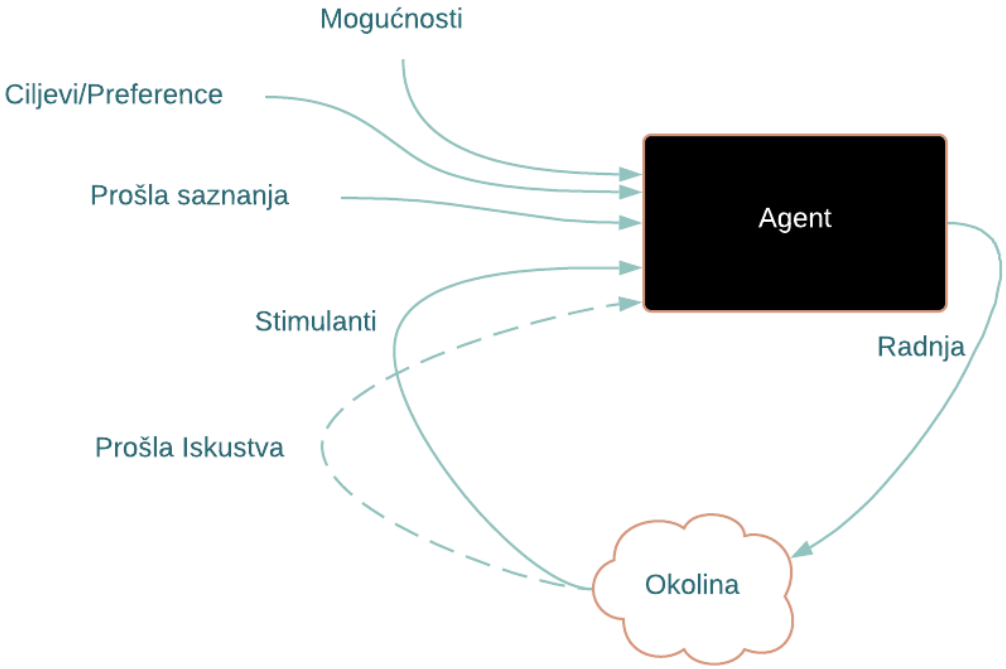
- "Njegova se radnja može smatrati prikladnom obzirom na njegove okolnostima te ciljevima koje pokušava dostići,
- fleksibilan je na promjenjivost okoline i vlastitih ciljeva,
- uči na svojim iskustvima, te
- donosi odluku koja je ispravna na temelju prijašnjih iskustva. Agent ne može direktno promatrati svijet i sve varijable koje mogu utjecati na odluku. Kada se govori o *softwareu* može se zaključiti da se zbog ograničene memorije ne mogu uzeti svi vanjski parametri u obzir. Također je važno napomenuti da agent ima ograničeno vrijeme u kojem može donesti odluku."

Sad kada se vidi što agenta čini inteligentim, može se definirati što je primarni znanstveni cilj kod umjetne inteligencije tj. kod umjetnih inteligentih agenata. "Primarni znanstveni cilj kod umjetne inteligencije je razumijevanje principa koji čine inteligento ponašanje moguće u prirodnim ili umjetnim sustavima" [3, str 4.]. Ovo je ostvarivo pomoću [3, str 4.]:

- "temeljne analize prirodnih i umjetnih agenata;
- formuliranja i testiranja hipoteza o tome što je potrebno kako bi se konstruirali inteligenti agenti; te
- dizajniranja, građenja i eksperimentiranja s računalnim sustavima koji izvršavaju zadatke za koje se smatra da je potrebna inteligencija."

Na slici 1 može se vidjeti agenta u interakciji s okolinom. Vidljivi su svi elementi koji su u ovoj cjelini definirali agenta. U svakom danom trenutku, što agent radi ovisi o [3, str 11.]:

- **prošlim saznanjima** o agentu i okolini u kojoj se nalazi
- **povijesti** interakcije s okolinom koja se zapravo sastoji od:
  - **stimulansa**/**podražaja** koji proizlaze iz trenutne okoline, što uključuje promatranje same okoline kao i okolnosti koja sama okolina nameće na agenta, te
  - **prošlih iskustva** ili drugih izvora relevantnih podataka od kojih se može nešto naučiti
- **ciljeva** koje agent želi postići ili **preferenca** na stanje svijeta
- **mogućnosti**, odnosno jednostavnih radnja koje je agent sposoban izvršiti.



Slika 1: Agent u interakciji s okolinom [3, str 11.]

2.2. Povijest umjetne inteligencije i Turingov test

Prva ideja za umjetnu inteligenciju može se pridodati kratkoj priči iz 1942. autora Isaaca Asimova zvana *Runaround* [4]. U toj priči on postavlja četiri zakona robotike:

0. Robot ne smije naškoditi čovječanstvu ili svojom pasivnošću dopustiti da se čovječanstvu naškodi.
1. Robot ne smije naškoditi čovjeku ili svojom pasivnošću dopustiti da se čovjeku naškodi, osim kad je to u suprotnosti s nultim zakonom.
2. Robot mora slušati ljudske naredbe, osim kad su one u suprotnosti s nultim ili prvim zakonom.
3. Robot treba štititi svoj integritet, osim kad je to u suprotnosti s nultim, prvim ili drugim. zakonom

Rad Isaaca Assimova inspirirao je generacije znanstvenika u polju robotike [4], a paralelno s time se događalo nešto strašno u svijetu - Drugi svjetski rat.

Ratovi uvijek donose destrukciju i štetu svim uključenim stranama, ali i nekima koje nisu direktno povezane, no ono što rat također donosi je ubrazni razvoj tehnologije. To se događa iz razloga što strana koja ima bolju tehnologiju povećava svoju šansu za pobjedu sa svojom superiornijom tehnologijom. Tako je prilikom drugog svjetskog rata naglo došla potreba za razvijanjem računala. To je bilo u svrhu balističkih projektila, ali i za dekodiranje tajnih njemačkih

šifri koje ni najbolji matematičari toga doba nisu mogli dekodirati. Alan Turing britanski je matematičar čiji je doprinos u znanosti računala dan danas evidentan. On je, između ostalog, sudjelovao u razbijanju njemačkod koda *"Enigma"* zajedno s Gordon Welchmanom [5].

No, Turing je i prije samog drugog svjetskog rata 1936. doprinjeo zajednici svojim hipotetskim računalom koje je kasnije prozvano "univerzalni Turingov stroj" (*eng. universal Turing machine*) na čijem se principu rodila ideja o pohranjivanju računalnih programa koje je John Von Neumann 1946. iskoristio za arhitekturu računala koja se dan danas koristi - Von Neumannova arhitektura [6].

Nakon samog drugog svjetskog rata, rast i razvoj računalnih tehnologija nije usporio te su se novi principi i modeli koji su i danas prisutni u računalnoj tehnologiji nastavili razvijati. Tako 1955. John McCarthy predstavlja pojam umjetne inteligencije [7].

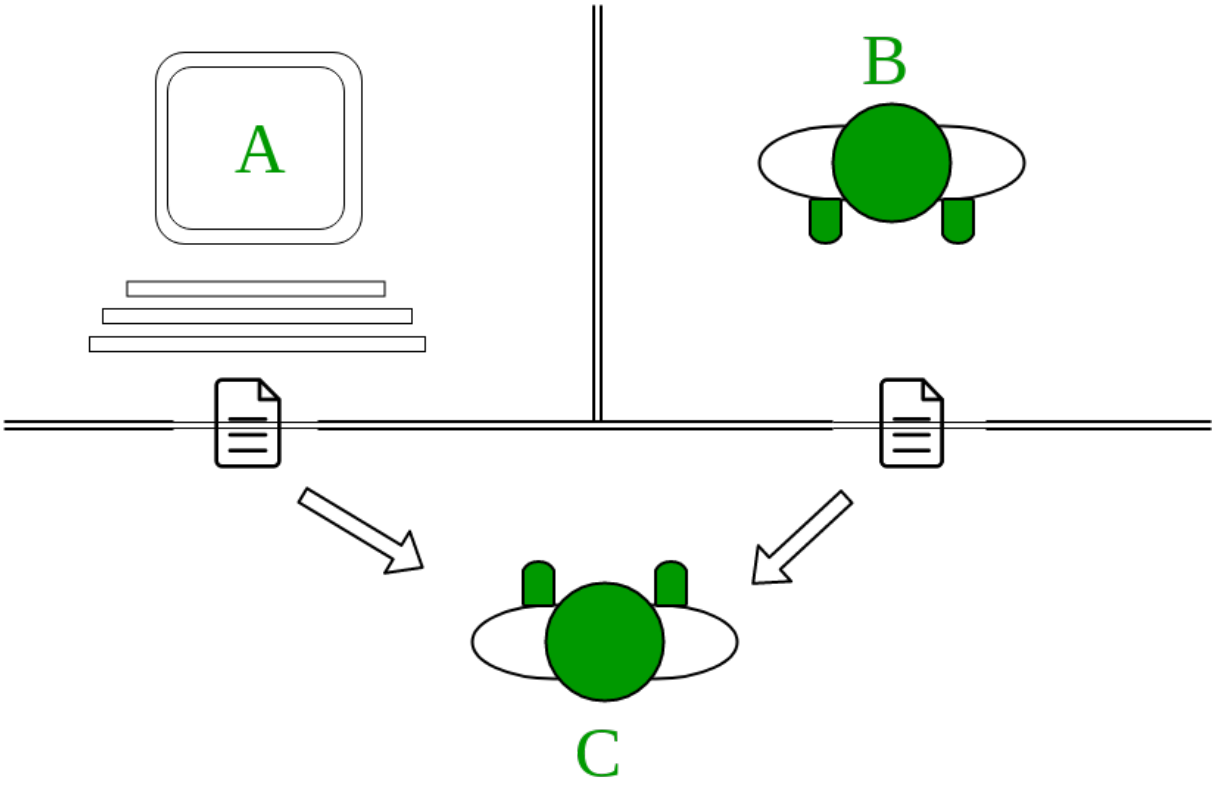
Izraz umjetna inteligencija prvi je put spomenuo John McCarthy 1956. godine [8, str 4.] kada je održao prvu akademsku konferenciju na tu temu. U to vrijeme tehnologija još nije bila ni blizu sposobnosti ogromne obrade podataka koja pružaju moderna računala, ali se na neka pitanja postavljena tada ni dan danas ne može dati cjeloviti odgovori.

Jedno od takvih pitanja je, može li zapravo računalo razmišljati odnoso razumijeti? Ovo pitanje postavio je Alan Turing 1950 [1]. Iako je filozofska priroda ovoga pitanja otvorena za raspravu, Turing je smislio način da se testira je li računalo inteligentno. Taj test se popularno zove Turingov test ili Igra imitacije. U vrijeme Alana Turinga i Turingovog testa je i sama ideja *chabtot*-a popularizirana. Važno je napomenuti da se ovaj test dan danas koristi kako bi se testirala inteligencija *chatbot*-a što je tema ovog rada.

Turingovom testu pristupaju sudac, *chatbot* i čovjek. Sudac je zadužen za postavljanje pitanja *chatbotu* i čovjeku preko poruka te pokušava utvrditi tko je od sugovornika *chatbot*, a tko čovjek. Cilj navedenog *chatbota* u ovom testu je održavati razgovor koji se ne razlikuje od ljudskog razgovora i „prevariti“ suca. Ako *chatbot* uspije uvjeriti suca da je ljudsko biće, on prolazi Turingov test te se smatra inteligentnim. Turing je svoj članak objavio 1950. [8, str 5.], ali tek se 1995. test počeo formalno provoditi. Kako bi *chatbot* uspješno mogao proći Turingov test, on mora biti dobar u sljedećem [9]:

- procesiranje prirodnog jezika
- reprezentacija znanja
- automatsko rasuđivanje
- strojno učenje

Slika 2 prikazuje standardnu interpretaciju Turingovog testa gdje igrač C (ispitivač) pokušava utvrditi koji igrač od igrača A i B je računalo, a koji je čovjek. Za takav ispit ispitivač dobiva na raspolaganje odgovore na pisana pitanja koja dobiva od igrača A i B. [2] Na temelju odgovora koje igrači A i B pružaju, Igrač C pokušava prepoznati *chatbota*. Ukoliko igrač C uspije utvrditi koji je igrač računalo, *chatbot* se ne smarta dovoljno inteligentim. U suprotnom



Slika 2: Prikaz Turingovog testa [2]

slučaju *chatbot* je prošao Turingov test te se smatra inteligentim. Ovo je jedan od jednostavnijih prikaza načina rada Turingovog testa.

3. Agent za razgovor

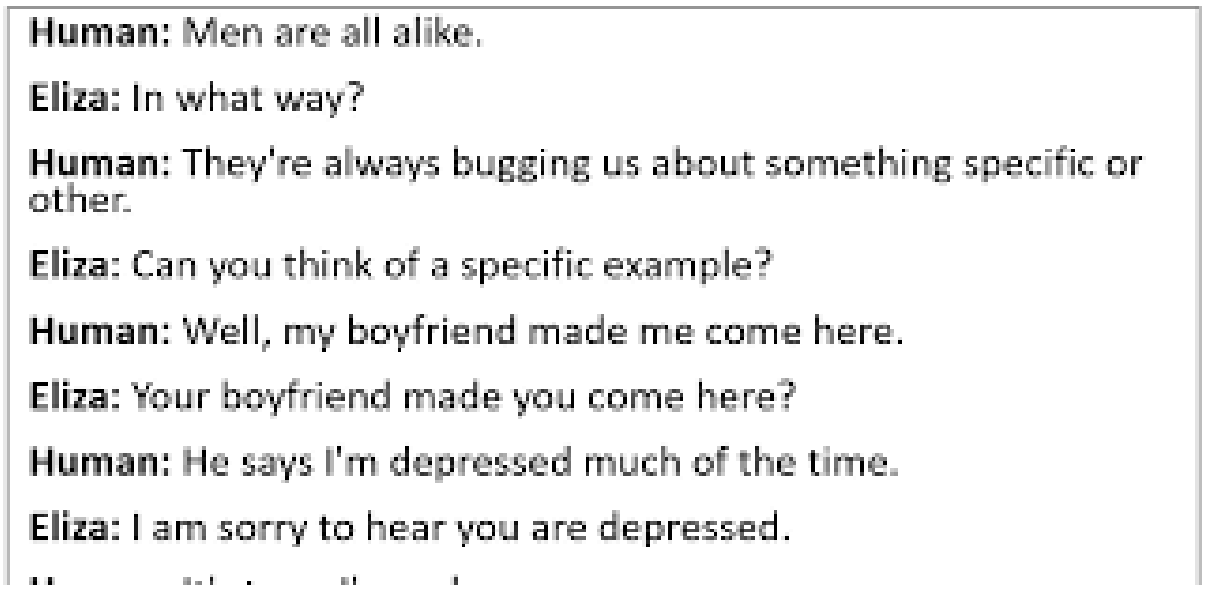
Kako bi se u potpunosti razumjeli *chatbot*-i i kako je došlo do moderne uporabe, svestranosti i koristi ove tehnologije, valja proučiti razvoj samih *chatbota*. Inspirirani Turingovim testom znastveni istraživači i inžinjeri razvijaju mnoge računalne sustave za konverzaciju [10]. Ovi rani programski sustavi su dizajnirani s namjerom prolaska Turingovog testa. Ideja je bila ovim računalnim programima simulirati način na koji se ljudsko biće ponaša u razgovoru s partnerom, koristeći pritom tekstualne i audio poruke. Ovi rani programi su zapravo prve inačice onoga što danas smatramo *chatbotom*, a prvi takav *chatbot* bio je nazvan Eliza.

3.1. Agenti za razgovor kroz povijest

3.1.1. Eliza

Eliza je prvi *chabot* koji je bio dostupan javnosti, a napravio ga je Joseph Weizenbaum 1966. godine na sveučilištu MIT [10]. Weizenbaum je sam napisao skripte po kojima Eliza komunicira s ljudskim korisnicima, pritom simulirajući Rogerijansku terapiju. To je terapija koja u fokus stavlja svog klijenta s nedirektnim pristupom i empatijom prema istom u svrhu motiviranja i poboljšanja njegovog emocionalnog stanja [11].

Ovaj chatbot je prihvaćao samo tekstualni input te nije "razumio" razgovor, već je tražio prikladne odgovore na temelju svojeg prepoznavanja uzorka (*eng. pattern matching*) [10]. Iako je Eliza bila ograničena svojim znanjem i domenom korisnika koji je odgovarao Elizi, *chatbot* se može smatrati uspjehom jer su mnogi sugovornici Elize imali osjećaj da pričaju s pravom osobom [10].



Slika 3: Prikaz interakcije korisnika i Elize [10]

Slika 3 prikazuje razgovor Elize s korisnikom. U ovom se primjeru može vidjeti kako

Eliza koristi svoje skripte kako bi simulirala Rogerijansku terapiju s pitanjima o specifičnosti korisnikove situacije. Upravo ovakvi odgovori i pitanja su ono što korisnicima daje osjećaj kao da pričaju s pravom osobom, vjerojatno zato što je većina ljudi kojima treba neka vrsta terapije povučena te jedan program gdje mogu staviti svoje misli na papir i prikladni odgovori Elize daju korisnicima osjećaj sigurnosti i te na neki neki način mogu ispuniti ljudsku potrebu za komuniciranjem ili barem da osjećaj da su saslušani.

No, Eliza nije prvi *chatbot* koji je uspio proći Turingov test, ta titula pripada *chatbotu* zvanom Parry [10].

3.1.2. Parry

Ideja za dizajn *chatbota* Parry bila je oponašanje paranoične osobe. Ovog *chatbot*-a napravio je Kenneth Colby 1975. Ovo ponašanje paranoične osobe je vjerojatno ključ koji je omogućio da Parry prođe Turingov test. Parry je i dalje poput Elize strukturiran na pristupu temeljenom na pravilima [10]. Ovaj pristup, kao i drugi, pristupi je detaljnije razrađen u sljedećim poglavljima. Parry je, doduše, imao bolju strukturu, veće sposobnosti "razumijevanja" jezika te nadodani mentalni model koji je simulirao emocije Parry-a. Kao primjer Parry reagira neprijateljski nastrojeno u slučaju da mu je razina ljutnje visoka.

Zanimljivo je što su Eliza i Parry 1972. preko Arpaneta bili postavljeni da međusobno razgovaraju [12]. Ovaj razgovor vidljiv je na slici 4. Valja napomenuti kako se u ovom razgovoru vidi razlika u pristupu, kao što je opisano u ovom poglavlju. Eliza je "empatična" i pokušava pomoći Parry-u dok se i Parry i dalje ponaša paranoično.

3.1.3. Alice

Alice je zadnji bot koji je istaknut u ovom radu za ovaj period. Alice je akronim *Artifical Linguistic Internet Computer Entity* što bi na hrvatskom slobodno prevedeno bilo lingvistički umjetni online inteligentni agent za . Razvio ju je Richard Wallae 1999. [10]. Koristi tzv. *Artificial Intelligence Markup Langugage* ili AIML ukratko. O samoj AIML tehnologij moguće je pronaći u 4.2.2. poglavlju uvoga rada. Alice je istaknuta upravo zbog svoje AIML tehnologije koja joj pojednostavljuje jezik te je osvojila Loebnerovu nagradu tri puta[10]. No, i Alice je imala svojih poteškoća, prvenstveno s održavanjem konverzacije na duži period vremena [10].

3.1.4. Inteligentni osobni asistenti

Iako inteligenti osobni asistent ne zvuči poznato većini ljudi, mnogi ih koriste, neki svakodnevno. Tim ljudima će asistent tvrtke Apple, zvana Siri, zvučati jako poznato. Osim Siri, i drugi veliki tehnološki divovi također su plasirali svoje asistente na tražište. Tu ubrajamo: Cortanu od Microsofta, Googleov osobni asistent i Alexu od Amazona. Ovi asistenti se mogu smatrati chatbotovima jer mogu komunicirati sa korisnikom te reagirati na njegove zahtjeve ili upite.

ELIZA: Good Evening. Tell me your problems. Please terminate input with a period or a question mark.

PARRY: People get on my nerves sometimes.

ELIZA: I am not sure I understand you fully.

PARRY: You should pay more attention.

ELIZA: Suppose you should pay more attention.

PARRY: You're entitled to your own opinion.

ELIZA: What makes you think I am entitled to my own opinion?

PARRY: It bothers me just to be around people in general.

ELIZA: Please go on.

PARRY: I went to the track at Bay Meadows a while back.

ELIZA: What does that suggest to you?

PARRY: I went to the races.

.....

ELIZA: What answer would please you most?

PARRY: It's not a matter or like or dislike.

ELIZA: What else would you like to discuss?

PARRY: I am getting tired of answering the same questions over and over.

Slika 4: Razgovor Elize i Parry-a [12]

Ovi agenti integriraju informacije od mnogobrojnih senzora uređaja u kojima se nalaze [10]. Neki od tih senzora uključuju lokaciju, vrijeme, pokrete detektirane žiroskopom, dodir, geste i sl. Osim što imaju više senzora, imaju i pristup mnogim izvorima podataka te mogućnost upravljanja ili podešavanja nekih alata ili aplikacija. Na primjer, osobni asistent može pretražiti internet, podesiti događaj u kalendar, namjestiti budilicu, pustiti muziku i slično. Ali osim ovih jednostavnijih mogućnosti, ovi asistenti mogu napraviti online rezervacije poput hotela te podsjetiti sve korisnike uključene u neki događaj na taj događaj.

Može se vidjeti kako određenim korisnicima ove funkcionalnosti mogu uvelike olakšati svakidašnji život. Ono za što je nekad trebala tajnica danas može napraviti mobilni uređaj s

nekoliko jednostavnih glasovnih komandi ili korisnikovim unosom.

3.1.5. Agenti za razgovor na društvenim mrežama

U današnje vrijeme društvene mreže postižu sve veću popularnost zbog dostupnosti i pristupačnosti pametnih telefona. Popularnost društvenih mreža je dodatno potaknuta konstatnim razvojem mobilnih tehnologija i mreža te samim time dolazi do potrebe za alternetivnim načinima komunikacije [10]. Pošto su ljudi digitalno povezani na internet, nije iznenađujuće da se javlja i veća potreba za *chatbotovima* jer velika količina ljudi s različitim potrebama provodi svoje vrijeme online.

Razlika između ovih *chatbota* i generacije prije je što su ovi *chatboti* specifično dizajnirani da budu od koristi za njihove korisnike, a ne da prođu Turingov test [10]. Naravno, s ovakvim pristupom bilo bi poželjno kad bi ovi *chatbotovi* mogli prepoznati emocije njihovih korisnika te prikladno reagirati. Ono što ih čini različitim od asistenata je što asistenti pokušavaju držati razgovor što kraćim te se fokusirati na jednostavno i učinkovito izvođenje zadatka. S druge strane, socijalni *chatboti* žele zadržati korisnikovu pažnju što dulje. Kako bi komunicirali kao ljudi, daju neke različite perspektive te daju nove teme kako bi razgovor što dulje trajao ovisno o osobnosti *chatbota* [10].

Tablica 1 ukratko pokazuje glavne karakteristike *chatbota* koji su spomenuti u ovom radu, ali i *DARPA Communicator Program* iz 2000. i Xiaolce koji predstavlja socijalnog *chatbota*, svaki ima svoje karakteristike vrijedne isticanja koje se mogu naći u tablici 1.

3.2. Tehnologija

U svrhu ovog rada istaknuto je nekoliko tehnologija, koncepata i okvira po kojima razvojni programeri razvijaju *chatbotove*.

3.2.1. Domena znanja

Prije svega, prilikom kreiranja *chatbota* potrebno je razmisliti o domeni znanja u kojoj će *chatbot* operirati. U svrhu ovoga imamo dvije opcije [13]:

- Otvorena domena znanja
- Zatvorena/ograničena domena znanja

Zatvorena domena znanja ograničava *chatbota* na određene ključne riječi ili namjere kako bi izvršio određeni zadatak. *Chatbotovi* otvorene domene, naprotiv, mogu ući s korisnikom u interakciju neovisno o temi [13]. U tablici 1 su tako Siri i Xialoce *chatbotovi* otvorene domene dok su prijašnji bili zatvorene.

U slučaju poduzeća koji gleda napraviti od *chatbota* neku vrstu korisničke službe prigodnija je zatvorena domena znanja. Prednost *chatbota* za poduzeće koji je zatvorene domene je

što bi se hipotetski trebao lakše moći izraditi. To je iz razloga što samo ograničavanje domene i postavljanje pravila i logike je u "prirodi" samih računala. Zbog te "prirode" razvojni programeri mogu dobro definirati pravila i stabla odluke *chatbot*-a.

S druge strane, otvorena domena znanja je pogodnija upravo kada je *chatbot* namijenjen da na razne načine komunicira s korisnikom. Ovo se temelji na nekim kompleksnijim okvirima te je mnogo kompliciranije za izvesti. Ovdje do izražaja dolazi strojno učenje s velikom količinom podataka kroz koje *chatbot* može učiti. Glavna mana ovih *chatbota* je što na pitanja otvorenog tipa znaju dati odgovore koji nikako nisu relevantni s trenutnim razgovorom ili promjene temu u neprikladnom trenutku.

U svrhu praktičnog dijela ovog rada implementiran je *chatbot* zatvorene domene znanja kako bi primjer bio jednostavniji. Nije potrebna otvorena domena za *chatbot*-a koji će služiti kao posrednik između čovjeka i izvora podataka jer mu je to primarna uloga.

3.2.2. Pristupi

U ovoj sekciji obrađeno je nekoliko pristupa koji se mogu koristiti pri izradi *chatbota*. Prvi pristup i najjednostavniji je pristup temeljen na pravilima.

Pristup temeljen na pravilima je pristup u kojem *chatbot* obzirom na pravila na kojima je treniran odgovara na upite[14]. Ovim pristupom nemoguće je izraditi *chatbot* koji ima otvorenu domenu znanja. To proizlazi iz razloga što je nemoguće postaviti sva pravila za otvorenu domenu. Otvorena domena se može usporediti sa svemirom, kako se svemir širi tako i otvorena domena uvijek raste te samim time ne možemo definirati sva pravila. Još jedna mana ovog simplističnog pristupa je i to što, ako upit koji korisnik zada ne odgovara pravilima na kojima je *chatbot* građen, sustav neće znati kako odgovoriti [14].

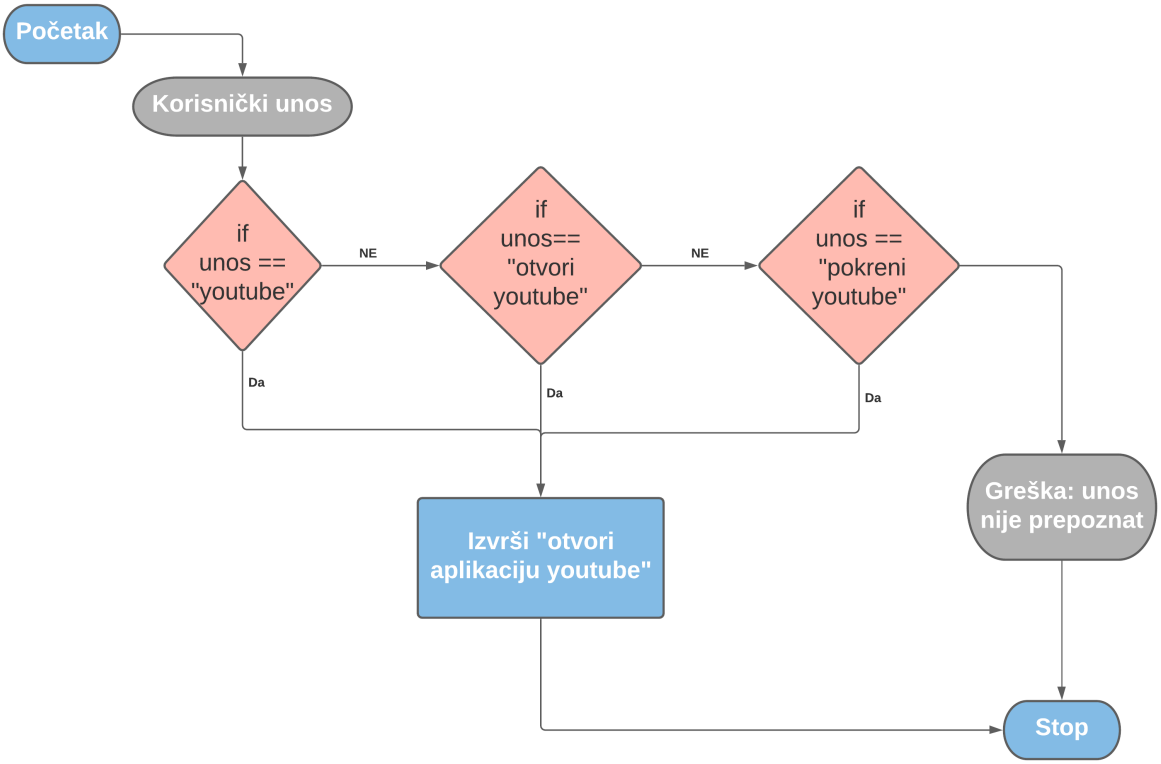
Slika 5 predstavlja dijagram toka jednostavnog pristupa prema pravilima. Obzirom na korisnički unos, *chatbot* prepznaje samo tri unosa koji onda otvaraju aplikaciju YouTube.

Kako bi razvojeni programeri malo proširili mogućnosti ovakvog *chatbota* kako ne bi sve bilo tako rigorozno, koriste AIML (*eng. Artificial Intelligence Markup Language*) za pisanje pravila *chatbot* sustava [14].

AIML (*eng. Artificial Intelligence Modelling Language*) je, kako navode Shahriare i sur. [15, str 87.], jezik za označavanje koji je baziran na *Extensible Markup Language* (XML)-u. Najpoznatiji *chatbot* baziran na AIML-u je Alice. AIML se sastoji od klase objekta podataka, odnosno AIML objekta koji opisuju ponašanje računalnih programa. Sastoji se od jedinica koje zovemo teme i kategorije. Kategorije su osnovna jedinica znanja iz AIML-a. "Svaka kategorija sastoji se od uzorka koji sadrži ulaze te predloška koji sadrži odgovor chatbota" [15, str 87.].

"AIML se sastoji od riječi, simbola sa povlakama te zamjenskih znakova simbol poput _ ili *. Velika i mala slova nisu bitna. Postoje tri vrste kategorija" [15, str 87.]:

- Atomske kategorije; Ova kategorija su uzorci koji nemaju zamjenske znakove.
- Zadane kategorije; Ove kategorije imaju neke zamjenske znakove poput *.



Slika 5: Jednostavan dijagram toka za pristup temeljen na pravilima

- Rekurzivne kategorije; To su kategorije koji imaju oznake <srai> i <sr> na koje se odnosi rekurzivna i simbolična redukcija.

    1. Simbolička redukcija: Smanjuje kompleks gramatičkih oblika u jednostavnije.
    2. Podijeli i ovladaj: dijeli ulaz u dva ili više dijelova i kombinira odgovor na svaku.
    3. Sinonimi: Vraća odgovor u uzorku najsličniji korisničkom uzorku.

AIML omogućuje kreiranje sučelja za korisnike s implementacijom koja je jednostavna za programirati, jednostavna za razumijeti i lagana za održati [16].

Drugi mogući pristup su *Self-learing Chatbots*, *chatbotovi* koji uče na prijašnjim interakcijama. Ovi botovi svoje znanje proširuju pomoću strojnog učenja. Postoje dva tipa ovog bota. [14].

- *chatbot* koji koristi modele temeljene na pretraživanju upita
- *chatbot* koji koristi generirajuće modele

Chatbotovi koji koriste modele na temelju pretraživanja upita sadržavaju mnoge upite te moguće permutacije odgovora. Za svaki upit *chatbot*, pretražuje najprikladnije odgovore iz

skupa mogućih permutacija odgovora. U ovom slučaju, jezik i struktura rečenica ne predstavljaju problem jer su prikladni odgovori unaprijed odlučeni te ne mogu ispasti neprikladni ako gledamo sa stajališta strukture rečenica [14].

S druge strane postoje *chatbotovi* koji koriste generirajuće modele koji ne odgovaraju na upite s istim skupom odgovora [14]. Naime, ovi *chatbotovi* gledaju riječ po riječ prilikom slaganja odgovora. Iz ovog razloga česte su jezične greške. Pogotovo ako se razmatra isti sa strane hrvatskog jezika koji je teško prenijeti u računalni oblik zbog padeža, lica i vremena.

Na ovoj vrsti *chatbota* potrebno je još raditi, jer *chatbot* koji je sklon greškama može iziritirati korisnika. Iz tog razloga poduzeća i dalje više preferiraju *chatbote* koji su temeljeni na pravilima [14]. *Chatbot* koji je implementiran u praktičnom primjeru također je *chatbot* definiran na pravilima.

Tablica 1: Sažetak chatbotova kroz povijest [10]

	ELIZA	PARRY	Alice	DARPA	Siri	Xialoce
Vrijeme nastanka	1966	1972	1995	2000	2011	2014
Skalabilnost	nema	nema	Prilagodba skripti	Ograničena	Skalabilan	Skalabilan
Istaknuta svojstva	Oponašanje ljudskog ponašanja u konverzaciji	Generira emocijonalni (ljuti) odgovr	Laka prilagodba kroz AIML	Razumi jezik i upravlja s dialogom. Orijentiran na ciljeve	Osobna digitalna asistencija korisnika	Gradi emocionalnu povezanost svojih korisnika.
Dostignuća	Prvi *chatboot*	Prošao Turingov Test	Osvojio 3 Loebnerove Nagrade	Shvaća zahtjeve na prirodnom jeziku i radi zadatke	Prvi Inteligenti Osobni Asistent	Prvi masovno raširen *chatbot*. Održavao TV program, izdao knjigu poezija
Vrsta ulaza	Tekst	Tekst	Tekst	Tekst i Glas	Tekst, Slika, Glas	Tekst, Slika, Glas
Struktura	Pristup na pravilima	Pristup na pravilima	Pristup na pravilima	Bazirano na učenju	Bazirano na učenju	Bazirano na učenju
Domena	Ograničena	Ograničena	Ograničena	Ograničena	Otvorena	Otvorena
Ključni tehnički napretci	Skripte, Prepoznavanje uzorka na temelju ključnih riječi, odgovori temeljeni na pravilima	Dodane osobnosti u svojim odgovorima	Korištenje AIML tehnologije	Statistički modela za razumijevanje govornog jezika i upravljanje dialogom	Reaktivna i proaktivna asistencija za različite domene	EQ za stvaranje emocionalne veze s korisnikom
Ključna tehnički limiti	Ograničena domena znanja	Ograničena	Veličina skripti može biti neprakticno velika	Funkcionira samo u domenama s dobro definiranim shemama	Nedostatak emocionalnog angažmana korisnika	osobnosti postanu nekonzistente u dužim razgovorima

4. Praktičan dio - integracija *chatbota* u aplikaciju Telegram

U svrhu praktičnog primjera napravljen je *chatbot* u aplikaciji Telegram. Programski jezik u kojem je pisan kod za *chatbot* je Python. *Chatbom* je nazvan Corona Botić te korisnik može nekoliko komandi unijeti kako bi dobio relevantne statistike o SARS-CoV-2 virusu. Ovo je jednostavan prikaz kako umjetni inteligentni agent može služiti kao posrednik između čovjeka i nekog izvora podataka.

Kako bi se dobili relevantni podaci o COVID-19 koristi se disease.sh API, što je za ovaj primjer izvor podataka. Prema IBM-u API *eng. Application Programming Interface* je "skup definiranih pravila koji objašnjavaju kako aplikacije i računala međusobno komuniciraju. API se nalazi između aplikacije i web servera, te je namijenjen da bude posredni sloj koji obrađuje razmjenu podataka između sustava" [17]." .Disease.sh je otvoreni API koji sadrži detaljne informacije o mnogobrojnim virusima, uključujući i COIVD-19. Nudi podatke na globalnoj razini te ima i skoro sve države te pojedine veće gradove u svojoj bazi podataka. Glavni tim ovog projekta sastoji se od četvero ljudi iz četiri različite države [18]. Na ovom primjeru pokazano je kako se jednostavno može implemenitrati običan *chatbot* koji može donijeti veliku vrijednost s podacima koje pruža.

U sljedećim sekcijama objašnjeno je zašto su korišteni određeni alati.

4.1. Telegram

Za praktični primjer odabrana je platforma Telegram jer je *open source*, odnosno bilo tko s tehničkim znanjem može raditi promjene ili integrirati nešto novo. Telegram ima lako dostupan i besplatan API. Konkretnije ima dvije vrsta API-a koji korisnici mogu koristiti [19]:

- Bot API
- Telegram API skupa sa TDLib (*eng. Telegram Database Library*

Telegramov Bot API omogućuje korisnicima da integriraju svoje vlastite *chatbotove* u Telegramov sustav. Ovo je sučelje koje se koristi u praktičnom primjeru kao poveznica *chatbot*-a između krajnjeg korisnika i izvora podataka.

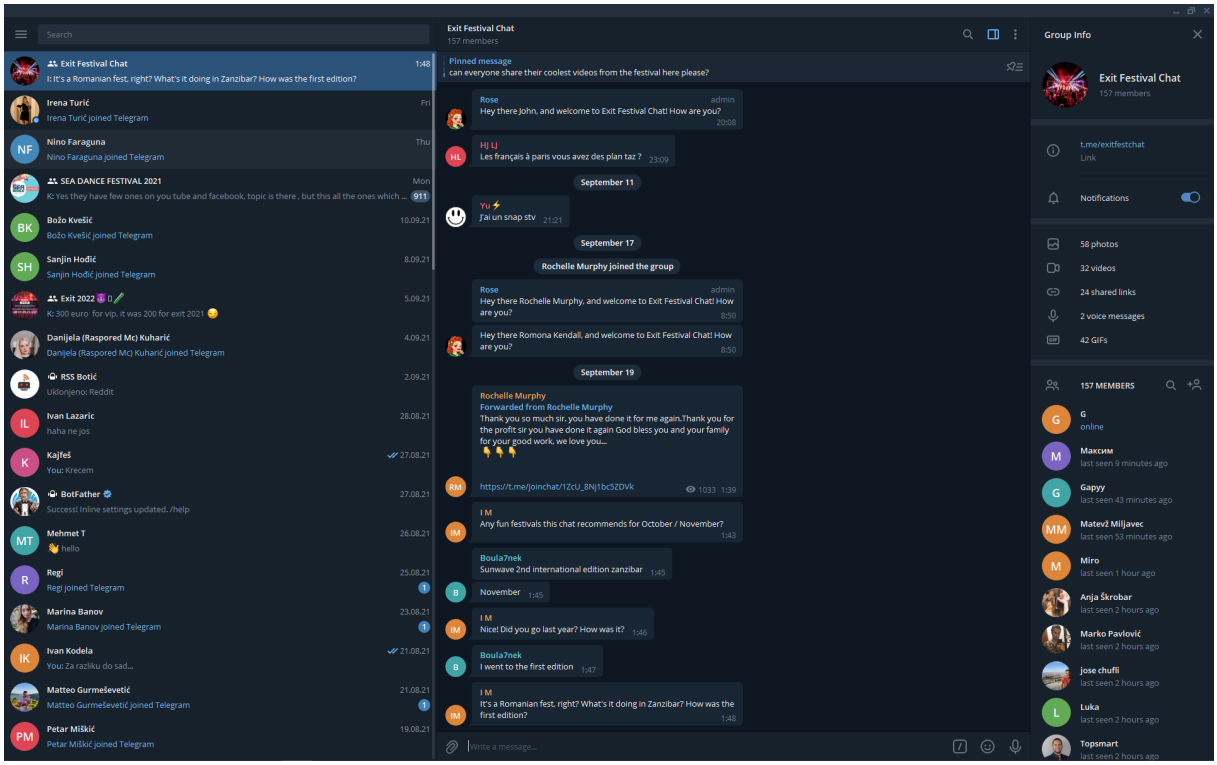
Telegram API služi samo za pristup servisima koje daje Telelgram, a TDLib služi kao okvir po kojem se mogu izraditi vlastite aplikacije sa strukturom Telegrama. To je moguće jer je Telegramov kod javno dostupan. Umjesto da razvojni programer krene raditi aplikaciju iz temelja, ovo uvelike pojednostavljuje cijeli proces.

Telegram se osim po svojim dostupnim API-jima razlikuje od konkurencije na slijedeće načine [20]:

- Jednostavnost - ukoliko je korisnik Telegrama ujedno bio i korisnik bilo kakve društvene mreže ili aplikacije za razmjenu poruka, Telegram izgleda intuitivno i poznato

te korisnik ne bi trebao imati problema sa samim korištenjem aplikacije.

- Privatnost - Ova aplikacija se ponosi sa svojom jakom ekripcijom te stavljaju privatnost svojih korisnika iznad svega. To ide do mjere da ni razvojni programeri ne mogu doći do podataka krajnjih korisnika. Telegramu je ovo omogućeno pomoću svojeg MTProto Protokola [21].
- Sinkronizacija - između mnogobrojnih uređaja razne vrste (osobno računalo, web servis, aplikacija za mobitel...)
- Brzina izvedbe - Telegram isporučuje svoje poruke brže od svojih konkurenata
- Neograničeni kapacitet za mediji i razgovore
- *Open source* - kao što je već navedeno svatko s tehničkim znanjem može napraviti izvedenice ove aplikacije ili je prilagoditi svojim potrebama
- Sigurnost - odnosi se na opasnost od hakerskih napada. Razvojni programeri Telegrama su toliko sigurni u programsko rješenje te nude i nagradu od 300,000$ bilo kome tko uspije doći do poruka korisnika Telegrama. [21].
- Socijalna platforma koja može u grupnim razgovorima sadržavati do 200,000 korisnika
- Dozvoljava korisnicima da se izraze u prilagodbi i izgledu same aplikacije



Slika 6: Izgled aplikacije Telegram

Prema službenoj stranici Telegrama, Telegramovi botovi mogu sljedeće [22]:

- Dobiti podešene novosti i obavijesti
- Integrati se s ostalim servisima - postoji bot za Gmail, YouTube, Wikipediju i sl.
- Ponuditi korisnika s plaćenom uslugom, uključujući i provođenje transakcije te se koristiti kao sučelje za trgovinu
- Kreirati alate
- Kreirati igricu za jednog igrača ili za više njih
- Kreirati socijalni servis - ovo može povezati ljude na temelju različitih parametra, poput zajedničih interesa ili međusobne udaljenosti.

Botovi na Telegram platformi se zapravo ponašaju kao posebni račun kojemu ne treba broj mobitela, kao što treba običnom korisniku. Korisnici mogu ući u interakciju s *chatbotom* ili preko skupa predefiniranih upita, ili kroz pisanje upita i poruka u chatu. Nemaju online status te su u korisničkom sučelju označeni s "bot". Također, ne mogu sami inicirati razgovor već korisnik prvo njih mora pretražiti i naći [22]. Slika 7 prikazuje cijeli proces izrade *chatbot*-a u razgovoru s BotFatherom. Naime, potrebno je samo unijeti komandu za novog bota (*/newbot*) te ga imenovati i dati mu korisničko ime. Valja napomenuti kako je API token pružan u zadnjoj poruci, ali je zamućen kako bi se zaštitio integritet bota.

Za svrhu implementacije *chatbota* korišten je programski jezik Python zbog jednostavnosti prikaza i samog koda kod integracije.

4.2. python-telegram-bot

Kako bi se sama integracija i manipulacija *chatbotom* dodatno pojednostavila, u svrhu ovog primjera korišten je omotač za Python programski jezik zvan python-telegram-bot. Ovo je javni repozitorij na githubu razvijen od mnogo programera koji postavlja okvire za manipulaciju s *chatbotom* [23]. Ovaj okvir zapravo služi kao sučelje za razvojne programere da pristupe gore navedenom Telegramovom bot API-u.

U svrhu ovog praktičnog primjera preuzet je ovoreni primjer python-telegram-bota zvan *echobot.py* koji je zapravo samo postavljen okvir koji prati kad mu korisnik pošalje poruku te mu vrati istu. [24].

4.3. Implementacija

Prije svega, prikazat ćemo korištene biblioteke te objasniti neke od njih:

Isječak kôda 1: Koristene biblioteke i resursi

```
1 import logging
2 import requests
3 import os
4 from telegram import Update, ForceReply
```

```
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters,
 CallbackContext
```

U isječku koda 1 najvažnije za osnovno funkcioniranje ovog *chatbota* su biblioteka *requests* koja omogućava slanje GET zahtjeva sa željenim API ključem te *telegram telegram.ext* koji proširuje python s funkcijama za lakše upravljanje *chatbotom*.

Prvo što se mora postaviti je poveznica s *chatbotom*. To se radi kroz biblioteku *os* koja sprema token od bota u varijablu za operacijski sustav da koristi kao na isječku koda 2. Osim tokena, postavljamo i identifikator razgovora *(eng. ChatID)* te varijablu *delay* koja će služiti kao odgoda prije no što bot ponovi neki upit. Treba napomenuti kako prikazani token ne predstavlja pravi token za *chatbota*.

Isječak kôda 2: Povezivanje pythona s *chatbotom*

```
if os.environ.get('TOKEN'):
    Token = os.environ['TOKEN']
    chatid = os.environ['CHATID']
    delay = int(os.environ['DELAY'])
else:
    Token = "1374271596:AAFpyT12E9FoJCt7UzolC6BHQQvfaoWgyRw"
    chatid = "python_telegram_project_mybot"
    delay = 60

if Token == "X":
    print("Token  nije postavljen!")
```

Nakon toga je potrebno definirati nekoliko komandi koje će korisnik može unijeti. Ovo su metode koje dohvaćaju i obrađuju podatke s odabranog API-ja te ih prilagođavaju za ispis. Za svrhu ovog rada prikazana je jedna metoda koja dohvaća podatke za statistiku u Hrvatskoj. Primjer se vidi u isječku koda 3.

Isječak kôda 3: Metoda za dohvat podataka o statistici COVID-19 virusa u Hrvatskoj

```
def cro_stat(update: Update, context: CallbackContext)->None:
    url="https://corona.lmao.ninja/v2/countries/croatia?yesterday=true&strict=true&
     query"
    data=requests.get(url).json()
    update.effective_message.reply_text(
        "\n\n SARS-CoV-2 (COVID-19) statistika za Hrvatsku" +
        "\n\n Ukupan broj zaraženih slučaja: "+str(data['cases'])+
        "\n\n Broj slučaja danas: "+str(data['todayCases'])+
        "\n\n Ukupan broj smrtnih slučaja: "+str(data['deaths'])+
        "\n\n Broj smrtnih slučaja danas: "+str(data['todayDeaths'])+
        "\n\n Ukupan broj oporavljenih: "+str(data['recovered'])+
        "\n\n Broj oporavljenih danas: "+str(data['todayRecovered'])+
        "\n\n Aktivni slučajei: "+str(data['active'])+
        "\n\n Kritični slučajevi: "+str(data['critical'])+
        "\n\n Broj slučaja na milion ljudi: "+str(data['casesPerOneMillion'])+
        "\n\n Broj smrtnih slučaja na milion ljudi: "+str(data['deathsPerOneMillion'
         ])+
        "\n\n Broj testova: "+str(data['tests'])+
        "\n\n Broj testova na milion ljudi: "+str(data['testsPerOneMillion'])+
```

```
18          "\n\n Broj aktivnih slučaja na milion ljudi: "+str(data['activePerOneMillion
             '])+
19          "\n\n Broj oporavljenih slučaja na milion ljudi: "+str(data['
             recoveredPerOneMillion'])+
20          "\n\n Broj kritičnih slučaja na milion ljudi: "+str(data['
             criticalPerOneMillion'])
21          )
```

Za svaki dohvat podataka s odabranog API-ja treba prvo pripremiti url željenog API-a. Vidimo da u urlu imamo parametar upita *croatia?* koji daje API-ju do znanja da treba dohvatiti podatke za Hrvatsku. Ove podatke spremamo u varijablu *data* u obliku JSON-a. Slanje GET parametra na API postiže se preko *request* biblioteke. Nakon toga *chatbot* šalje formatirani string kao poruku.

Po potrebi može se definirati još komandi, no sve ih je potrebno postaviti u glavnu funkciju gdje se naposlijetku upravlja s *chatbotom*. Ukoliko funkcija nije postavljena u glavnu funkciju, jednostavno *chatbot* ne će moći pristupiti željenoj funkciji.

Isječak kôda 4: Glavna funkcija

```
1  def main():
2      """Start the bot."""
3      # Napravi varijablu updater preko koje se upravlja botom.
4      updater = Updater(token=Token, use_context=True)
5      job_queue = updater.job_queue
6      dispatcher = updater.dispatcher
7
8      #postavi varijablu koja upravlja okidačima
9      dispatcher = updater.dispatcher
10
11     # dodaj komande koje korisnik može unijet
12     dispatcher.add_handler(CommandHandler("start", start))
13     dispatcher.add_handler(CommandHandler("help", help_command))
14     dispatcher.add_handler(CommandHandler("europe", list_country))
15     dispatcher.add_handler(CommandHandler("cro", cro_stat))
16
17
18     # na ostale komande koje se unose preko poruka
19     dispatcher.add_handler(MessageHandler(Filters.text & ~Filters.command, echo))
20
21
22     # Pokretanje bota
23     updater.start_polling()
24
25     updater.idle()
26
27
28 if __name__ == '__main__':
29     main()
```

Isječak koda 4 prikazuje glavnu funkciju. Istaknuta je varijabla *dispatcher* preko koje se postavljaju okidači koji pozivaju definirane funkcije kad korisnik unese "/*komanda*". Ova funkcija je

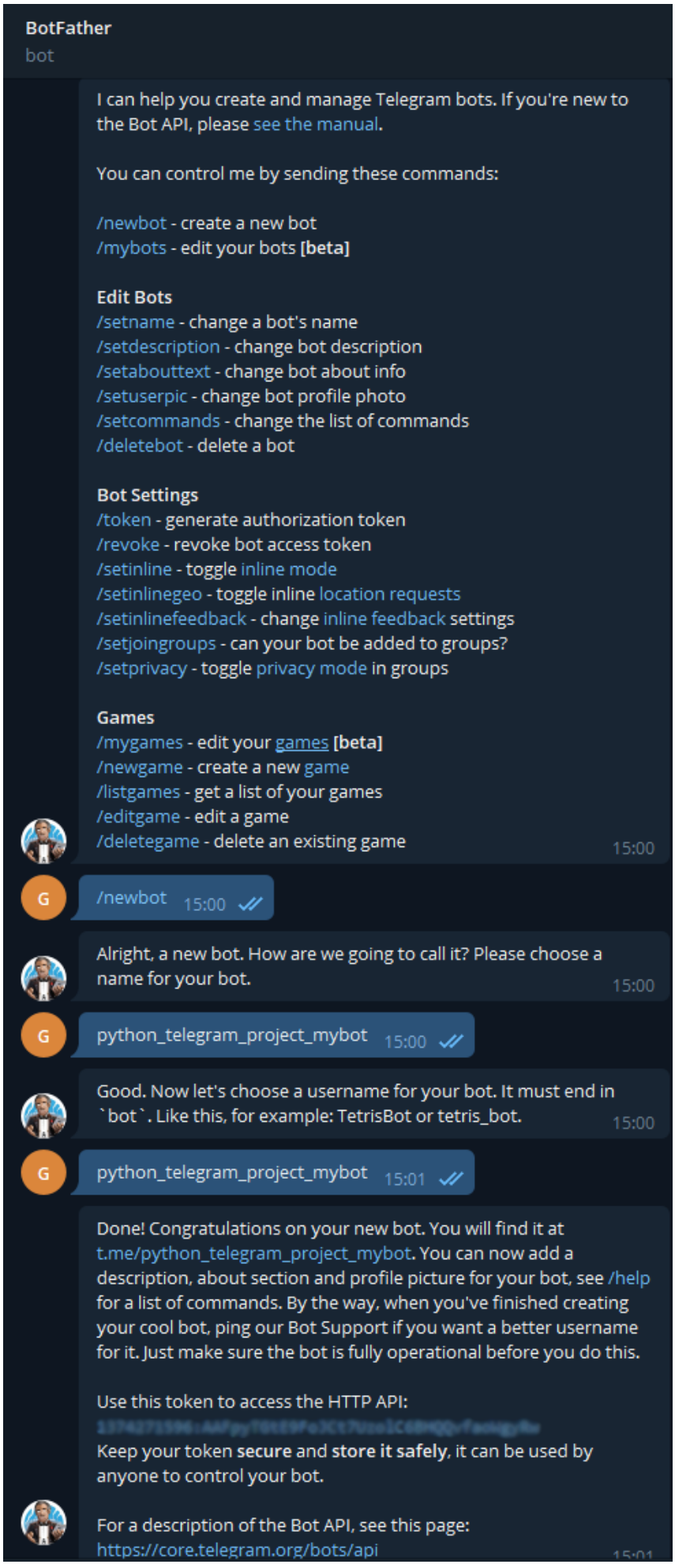
direktno preuzeta s izvora [24], samo su nadodane funkcije s *dispatcherom* sa željenim podacima iz API-ja i funkcije uključene za dohvaćanje tih podataka.

Ove osnovne stvari su sve što je potrebno kako bismo imali postavljenog *chatbot*-a. Naravno ovisno o potrebi za *chatbot*-om metode se mogu dodatno proširiti i dodati koliko je god funkcionalnosti poduzeću potrebno. Sljedeća sekcija prikazuje primjer korištenja ovog *chatbot*-a.

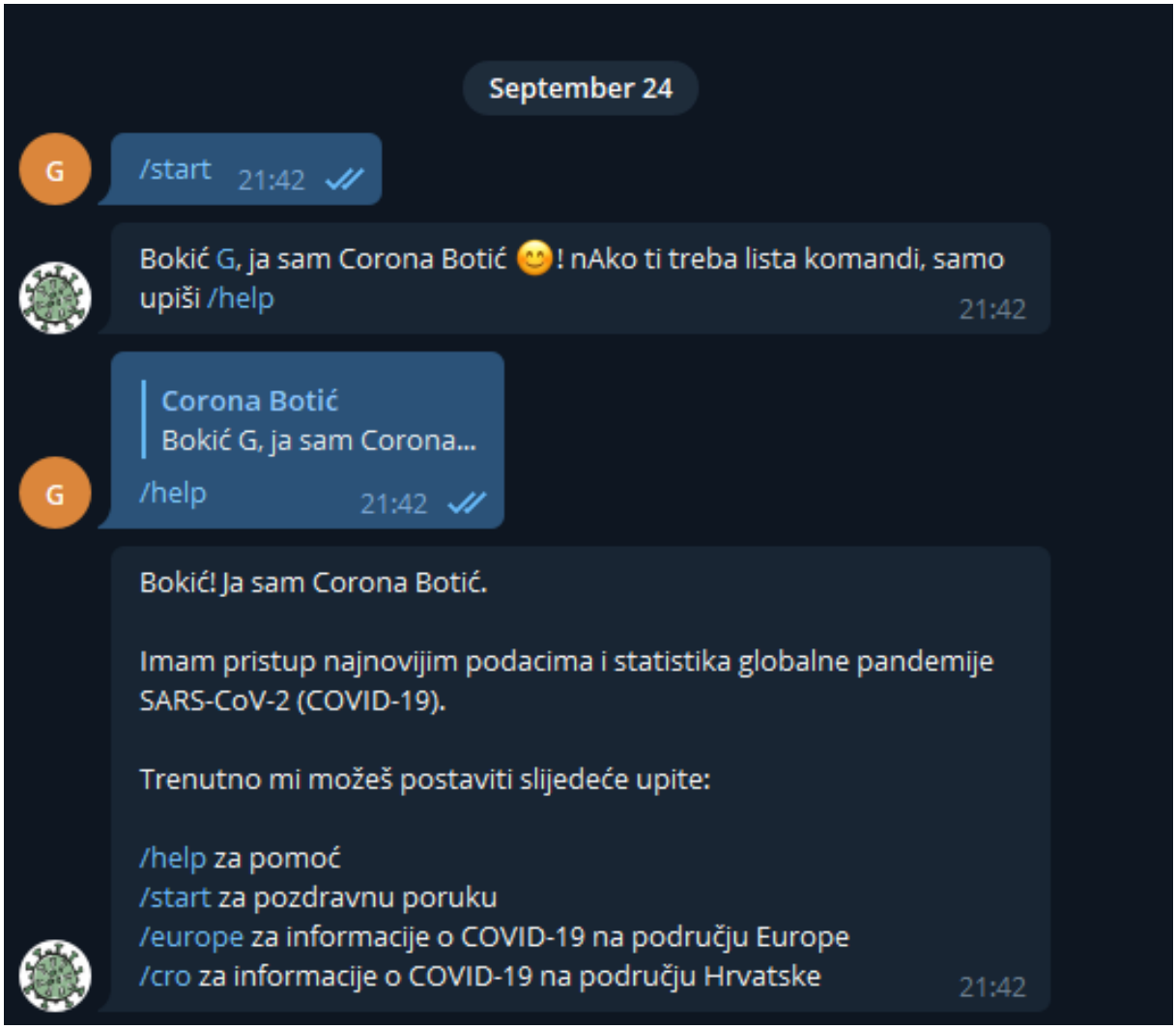
4.4. Primjer korištenja

Prva prikazana naredba je naredba "*/start*" zajedno s naredbom "*/help*". Primjer je prikazan na slici 8. Na slici 8, osim toga, vidi se kako *chatbot* ima implementirane dvije funkcije: */cro* i */europe*. */cro* funkcija vraća podatke za statistiku u Hrvatskoj, dok */europe* vraća podatke za statistiku na području cijele Europe.

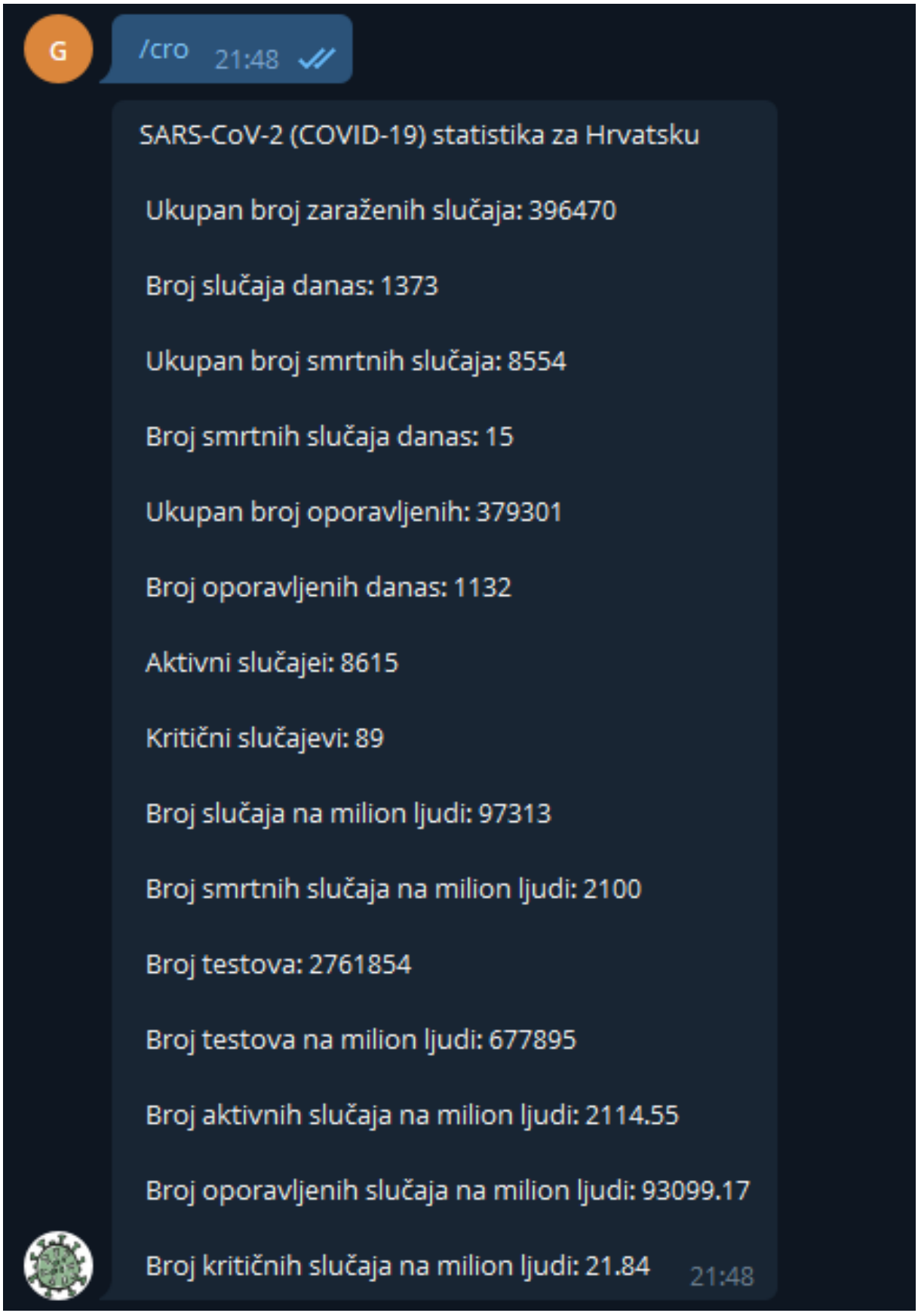
/cro funkcija daje izlaz prikazan na slici 9. Na toj se slici mogu vidjeti svi podaci koje *chatbot* može očitati. Naravno ovisno o potrebi za *chatbotom* mogu se filtrirati i sortirati podaci te se može nadodati nekakva dodatna logika. Funkcija *europe* daje identičan format ispisa, ali s podacima na području cijele Europe.



Slika 7: Razgovor s BotFather-om za kreiranje novog *chatbota*



Slika 8: Primjer početnog razgovora s *chatbot*-om



Slika 9: Ispis funkcije */cro*

5. Zaključak

Može se zaključiti da *chatbotovi* postaju sve češća pojava u današnje vrijeme te su se u određenim industrijama pokazali kao pogodni za što lakše poslovanje poduzeća. Njihova jednostavnost korištenja omogućila je mnogim poduzećima smanjenje radne snage te brže rješavanje problema i lakšu komunikaciju s korisnicima.

Prema vlastitom mišljenju negativna strana *chatbotova* je previsoka personalizacija *chatbotova* (*chatbot* može biti predstavljen kao ljudsko biće s imenom, godinama, osobinama i mnogim drugim karakteristikama) te se usamljeni korisnici mogu previše povezati s *chatbotom* te ga smatrati prijateljem.

U završnom radu se *chatbot* integrirao u aplikaciju Telegram kako bi korisniku mogao odgovarati na jednostavna pitanja s unaprijed pripremljenim odgovorima, ovdje se može vidjeti poveznicu *chatbota* kao posrednika između čovjeka i izvora podataka. Ovakva implementacija može biti korisna za brzu komunikaciju i postavljanje pitanja za koja su već unaprijed pripremljeni odgovori. Kroz izrađen vlastiti primjer se može vidjeti na jednostavan način kako rade *chatbotovi* te na koji način se mogu uvrstiti u poslovanje poduzeća.

Popis literature

[1] E. Adamopoulou i L. Moussiades, „An Overview of Chatbot Technology," *Artificial Intelligence Applications and Innovations*, I. Maglogiannis, L. Iliadis i E. Pimenidis, ur., Cham: Springer International Publishing, 2020., str. 373–383, ISBN: 978-3-030-49186-4.

[2] S. Bansal. (). „Turing Test in Artificial Intelligence," GeeksforGeeks, adresa: https://www.geeksforgeeks.org/turing-test-artificial-intelligence/ (pogledano 20. 8. 2021.).

[3] A. K. M. David L. Poole, *Artificial Intelligence: Foundations of Computational Agents*, Cambridge, 2010.

[4] M. Haenlein i A. Kaplan, „A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence," *California Management Review*, sv. 61, br. 4, str. 5–14, 2019. DOI: 10.1177/0008125619864925. eprint: https://doi.org/10.1177/0008125619864925.

[5] (). „Hoow Alan Turing cracked the Enigma code," adresa: https://www.iwm.org.uk/history/how-alan-turing-cracked-the-enigma-code (pogledano 20. 9. 2021.).

[6] M. Davis, „The Universal Computer: The Road from Leibniz to Turing," *The American Mathematical Monthly*, sv. 109, 6. 2002. DOI: 10.2307/2695463.

[7] (). „History of Artificial Intelligence – AI of the past, present and the future!" DataFlair, adresa: https://data-flair.training/blogs/history-of-artificial-intelligence/ (pogledano 23. 9. 2021.).

[8] C. Smith, *The History of Artificial Intelligence*. University of Washington, 2006., ISBN: 3-4.

[9] P. N. Stuart Russell, *Artificial Intelligence: A Modern Approach, 4th Edition*. Berkeley: University of California, 2020.

[10] H.-Y. Shum, X. He i D. Li, „From Eliza to XiaoIce: challenges and opportunities with social chatbots," *arXiv preprint arXiv:1801.01957*, 2018.

[11] (). „Person-Centered Therapy (Rogerian Therapy)," Good Therapy, adresa: https://www.goodtherapy.org/learn-about-therapy/types/person-centered (pogledano 20. 9. 2021.).

[12] M. Garber. (). „When PARRY Met ELIZA: A Ridiculous Chatbot Conversation From 1972," adresa: https://www.theatlantic.com/technology/archive/2014/06/when-parry-met-eliza-a-ridiculous-chatbot-conversation-from-1972/372428/.

[13] D. Adiwardana, M.-T. Luong, D. R. So, J. Hall, N. Fiedel, R. Thoppilan, Z. Yang, A. Kulshreshtha, G. Nemade, Y. Lu i Q. V. Le, „Towards a Human-like Open-Domain Chatbot," *CoRR*, sv. abs/2001.09977, 2020. arXiv: 2001.09977.

[14] Throat, S. A., Jadhav i Vishakha, „A Review on Implementation Issues of Rule-based Chatbot Systems," 2020.

[15] S.-A.-M. Md. Shahriare Satu Md. Hasnat Parvez, „Review of integrated applications with aiml based chatbot," *2015 International Conference on Computer and Information Engineering (ICCIE)*, Rajshahi, Bangladesh: IEEE, ISBN: 978-1-4673-8343-1.

[16] (). „AIML Tutorial," Tutorials Point, adresa: https://www.tutorialspoint.com/aiml/index.htm (pogledano 15. 9. 2021.).

[17] IBM. (2021.). „Application Programming Interface (API)," IBM, adresa: https://www.ibm.com/cloud/learn/api (pogledano 11. 10. 2021.).

[18] (). „disease.sh," adresa: https://github.com/disease-sh/API (pogledano 24. 9. 2021.).

[19] (). „Telegram APIs," adresa: https://telegram.org/api (pogledano 24. 9. 2021.).

[20] (). „Telegram Messenger," adresa: https://telegram.org/ (pogledano 24. 9. 2021.).

[21] (). „Telegram FAQ," adresa: https://telegram.org/faq#q-how-secure-is-telegram (pogledano 24. 9. 2021.).

[22] (). „Bots: An introduction for developers," adresa: https://core.telegram.org/bots (pogledano 24. 9. 2021.).

[23] (). „python-telegram-bot," adresa: https://github.com/python-telegram-bot/python-telegram-bot (pogledano 14. 9. 2021.).

[24] (). „echobot.py," adresa: https://github.com/python-telegram-bot/python-telegram-bot/blob/master/examples/echobot.py (pogledano 24. 9. 2021.).

Popis slika

1. Agent u interakciji s okolinom [3, str 11.] 4
2. Prikaz Turingovog testa [2] 6
3. Prikaz interakcije korisnika i Elize [10] 7
4. Razgovor Elize i Parry-a [12] 9
5. Jednostavan dijagram toka za pristup temeljen na pravilima 12
6. Izgled aplikacije Telegram 16
7. Razgovor s BotFather-om za kreiranje novog *chatbota* 21
8. Primjer početnog razgovora s *chatbot*-om 22
9. Ispis funkcije */cro* 23

Popis tablica

1. Sažetak chatbotova kroz povijest [10] 14

Popis isječaka koda

1. Koristene biblioteke i resursi 17
2. Povezivanje pythona s *chatbotom* 18
3. Metoda za dohvat podataka o statistici COVID-19 virusa u Hrvatskoj 18
4. Glavna funkcija 19