

Socialno inteligentni komunikator: razvojni prototip, demonstracijsko okolje in študijsko orodje

Emma Masterl, Samo Jean, Jure Potočnik, Urban Burnik, Andrej Košir

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana, Slovenija
E-pošta: andrej.kosir@fe.uni-lj.si

Povzetek. V delu predstavljamo načrtovanje, razvoj in implementacijo pogovornega sistema, ki ga imenujemo socialno inteligentni komunikator. Ciljev komunikatorja je več in zajemajo razvoj delujočega prototipa za razvoj realnih rešitev komunikacije s pametnimi sistemi, demonstracijsko okolje za prikaz naprednih tehnologij umetne socialne inteligence in hkrati študijsko orodje za poučevanje na univerzitetnem nivoju. Bistvo projekta je dejstvo, da je pri razvoju projekta dejavno sodelovalo več skupin študentov. Izhodiščne zahteve so vodile v vpeljavo virtualnih računalnikov, kjer smo se odločili za sodobno okolje Docker.

Ključne besede: pogovorni sistem, umetna socialna inteligenca

Socially intelligent communicator: prototype, demo environment and learning environment

In this paper, we present the design, development and implementation of a socially intelligent conversational system called a socially intelligent communicator. There are several goals of the communicator and they include the development of a working prototype for the development of real applications, an environment for demonstrating advanced technologies of artificial social intelligence and at the same time a learning and teaching tool at the university level. The essence of the project is the fact that groups of students contributed to the development of the project significantly. The initial requirements led to the introduction of virtual machines, where we opted for the modern Docker environment.

Keywords: conversational system, artificial social intelligence

1 UVOD

Razvoj socialno inteligentne interakcije med uporabnikom in pametnimi sistemi je izziv telekomunikacij bližnje prihodnosti. Zajema kombinacijo delno rešenih problemov (npr. razpoznavo govora), delno rešenih problemov (npr. situacijsko zavedanje) in problemov v začetni fazi razvoja (npr. razpoznavo socialnih signalov). Napredek posameznih področij, kot so razpoznavo in sinteza govora [1], [2], merjenje in analiza psiho-fizioloških signalov uporabnikov [3], [4], procesiranje naravnih jezikov s strojnimi učnjem, situacijsko zavedanje uporabnikov na podlagi interneta stvari in senzorjev ter spletnih storitev so skozi desetletja razvoja pogovorne sisteme privedli do uporabne oblike. Danes so pogovorni sistemi v redni množični rabi kot podpora strankam na različnih področjih, kot so bančne storitve.

Cilji izgradnje socialno inteligentnega komunikatorja so bili raznoliki in vključujejo izgradnjo uporabnega prototipa za razvoj aplikacij na področju podpore starejšim, demonstracijsko okolje ter okolje za izdelavo študentskih projektov pri predmetih Optimizacija v telekomunikacijah in Uporabniku prilagojena komunikacija.

Elementi socialno inteligentnega komunikatorja so razpoznavo in sinteza govora, razumevanje in generiranje naravnega govora, inteligenca pogovora, kontekstualizacija za situacijsko zavedanje, senzorji za analizo izmerjenih signalov za oceno socialnih signalov uporabnikov ter integracija socialnih signalov v inteligenco pogovora. Več o zgradbi predlaganega pogovornega sistema je zapisano v poglavju 3.

Izbran razvojni scenarij pogovora, na katerem je temeljil razvoj predlaganega komunikatorja in tudi nekatere razlage v tem članku, je svetovanje mobilnosti v Ljubljani. Ta vključuje izbiro med javnim prevozom LPP in pešačenjem. Relevantne situacije uporabnika so trenutni položaj uporabnika, cilj potovanja z razdaljo potovanja, položaj konkretnih avtobusov prevoznika LPP in vreme. Relevanten socialni signal uporabnika je hitenje, ki kaže, ali je čas potovanja za uporabnika bistven ali ne.

Umetna socialna inteligenca kot napredna tehnologija je ena od manjkajočih nadgradenj pogovornih sistemov, ki bi jih približala uporabi v naravnem okolju. Z razpoznavo, razumevanjem, sintezo in uporabo uporabnikovih socialnih signalov, kot so pozornost, angažiranje, stres, čustvena stanja, oklevanje in hitenje, pogovor s sistemom približa pogovoru med dvema osebamama.

Situacija uporabnika in situacijsko zavedanje, kot sta njegov cilj pogovora s sistemom in relevantna situacija okolja, sta ključna za izvajanje inteligentnega pogovora

v smeri doseganja ciljev pogovora. Primeri situacij so stanje pametnega doma, poznavanje dnevne rutine uporabnika, stanje povezanih sistemov, kot so javni prevozi, ipd. Tehnologije interneta stvari in senzorski sistemi omogočajo določanje situacije uporabnika v zelo različni meri, odvisno od dostopnosti podatkov in posebnosti situacij. Rešitve za določanje situacije so še vedno zelo specifične za posamezno tematiko pogovora. V primeru našega razvojnega scenarija sta to položaj avtobusov prevoznika LPP in vreme.

Cilj tega prispevka je predstavitev socialno inteligentnega komunikatorja in njegovega potenciala kot prototipnega in demonstracijskega okolja ter hkrati študijskega okolja za poučevanje napredne interakcije med uporabnikom in pametnimi sistemi.

V drugem poglavju predstavljamo področja socialno inteligentne komunikacije s pametnimi napravami, sledi predstavitev predlaganega komunikatorja z izhodiščnimi zahtevami, četrto poglavje pa podaja razvoj in uporabo komunikatorja z osnovnimi testiranjimi. Članek zaključujemo z zaključki in nadaljnjim delom.

2 PODROČJA SOCIALNO INTELIGENTNE KOMUNIKACIJE S PAMETNIMI NAPRAVAMI

V poglavju podajamo osnovne sestavne dele socialno inteligentne komunikacije.

2.1 Pogovorni sistemi

Pogovorni sistemi postajajo uporabna alternativa uporabniški interakciji človeka (uporabnika) s pametnimi sistemi [5]. Uporabni so le v specifičnih scenarijih, kjer interakcijo približajo naravni komunikaciji med ljudmi. Veliki ponudniki storitev ponujajo vsak svojo rešitev, kot so Google Assistant [6], Amazon Alexa [7], Microsoft Cortana [8] in Apple Siri [9].

Google Assistant je zasnovan tako, da vam omogoča pogovorne interakcije, potem ko izgovorite besede za bujenje »OK Google« ali »Hey Google«. Google lahko prepozna tudi glasovne profile za različne ljudi, tako da ve, kdo se z njim pogovarja, in lahko temu prilagodi odzive. Prav tako lahko zahtevate več stvari hkrati.

Amazonov pametni pomočnik je dostopen prek linije zvočnikov Amazon Echo ter Amazonove linije tabličnih računalnikov Fire in Fire TV. Od vseh pomočnikov AI je Alexa na voljo na najrazličnejših napravah, kar zadeva oblikovanje.

Microsoft Cortana je virtualni pomočnik, ki ga je razvil Microsoft in uporablja iskalnik Bing za izvajanje nalog, kot sta nastavitve opomnikov in odgovarjanje na vprašanja za uporabnika.

Siri je digitalni osebni pomočnik, integriran v operacijske sisteme naprav Apple, ki uporabnikom naprav Apple omogoča, da dobijo odgovore na vprašanja. Čeprav je na voljo več različnih ponudnikov na danem področju in ima vsak svojo prednost, je naša odločitev za agenta SECA temeljila na dejstvu, da je bila

to edina možnost, ki je tedaj ponujala vključitev slovenskega jezika.

2.2 Situacijsko zavedanje in kontekst

Zavedanje situacije je poleg inteligence dvogovora najpomembnejši faktor uspešnosti pogovornega sistema. Uspešnost določanja kontekstov (situacij) uporabnika je v največji meri odvisna od dostopnosti relevantnih kontekstnih informacij [10]–[12]. Kontekst ima drevesno strukturo, kjer so tri glavne veje *čas*, *položaj* in *aktivnost*. Nadaljnja razvejitev je odvisna od storitve in scenarija komunikacije, glejte podpoglavje 3.2.3.

2.3 Umetna socialna inteligenca

Umetna socialna inteligenca je sposobnost pametnega sistema, da razpozna uporabnikove socialne signale, jih uporabi v dobro uporabnika (storitev se glede na te signale spreminja) in socialne signale po potrebi tudi generira in sporoča. Socialni signali uporabnika so vse, kar ostane od njegove naravne komunikacije (govor), ko odstranimo vsebino govora. To so barva glasu, izraz na obrazu, čustvena stanja, pozornost, angažiranost itd. Posamezni elementi umetne socialne inteligence se razvijajo že več desetletij, pomemben prispevek sta članka [13] in [14]. Tehnologije razpoznave socialnih signalov naglo napredujejo [4], [15], [16], prav tako sinteza socialnih signalov [1], [17], [18].

Za vsako aplikacijo je glede na njene cilje treba izbrati relevanten socialni signal, za priporočilni sistem za filme smo npr. izbrali signal *oklevanje* [19]. Za tu predlagani komunikator smo izbrali signal hitenje.

3 SOCIALNO INTELIGENTNI KOMUNIKATOR: ZAHTEVE IN PREDLAGANA REŠITEV

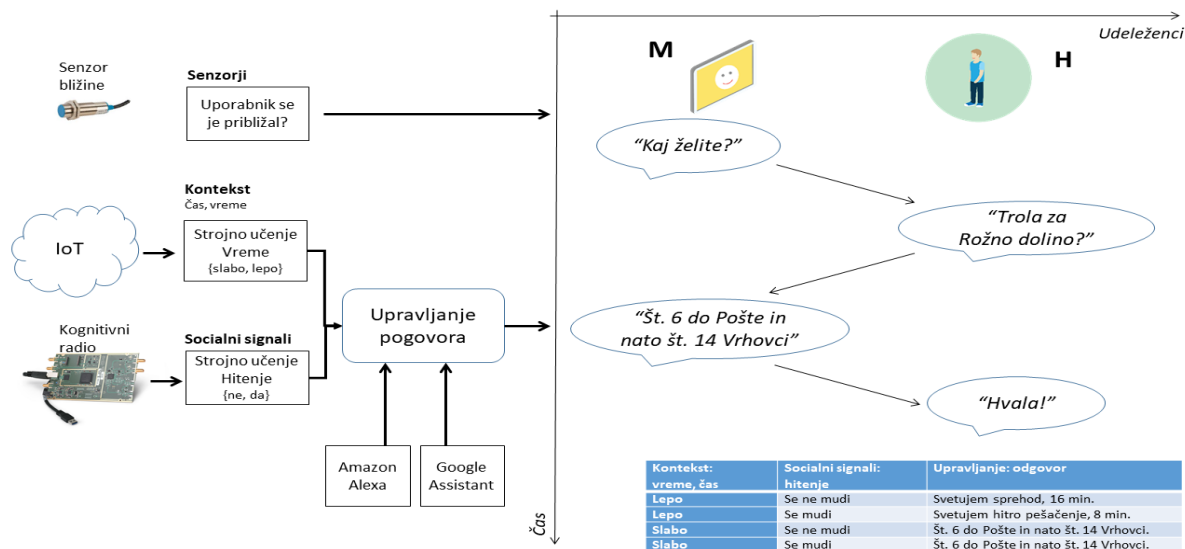
V poglavju predstavljamo izhodiščne zahteve in predlagano rešitev socialno inteligentnega komunikatorja.

3.1 Zahteve za sistemske rešitve

Izhodiščne zahteve za gradnjo socialno inteligentnega komunikatorja so:

1. Enostaven in ločljiv razvoj posameznih komponent komunikatorja, ki omogoča razvoj in testiranje na daljavo.
2. Prilagodljiva arhitektura za integracijo zunanjih funkcionalnosti (npr. spletne jezikovne storitve) in hkrati integracija lokalnih senzorjev.
3. Odprtokodna razvojna okolja in odprtokodne knjižnice zunanjih funkcionalnosti.
4. Sistematično testiranje in gradnja učnih množic za izboljševanje funkcionalnosti.
5. Učinkovita virtualizacija sistema za razvoj na daljavo.

Navedene zahteve omogočajo tudi uporabo sistema kot študijskega okolja.

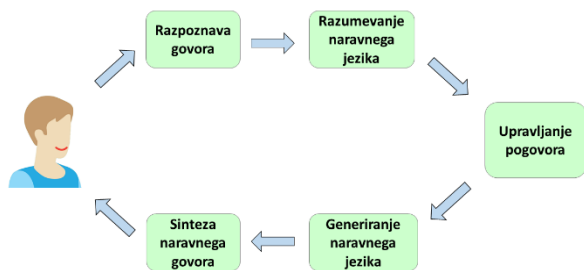


Slika 1 Scenarij pogovora z arhitekturo sistema kot izhodišče za razvoj.

Specifične zahteve izhajajo neposredno iz podprtih scenarijev. Izhodiščni scenarij za predstavljeni socialno inteligentni komunikator so na sliki 1.

3.2 Arhitektura in komponente socialno inteligentnega komunikatorja

V podglavju predstavljamo predlagano rešitev socialno inteligentnega komunikatorja, ki ustreza zgoraj navedenim zahtevam.



Slika 2. Komponente osnovnega pogovornega sistema, skupne vsem pogovornim sistemom. V predlaganem pogovornem sistemu smo dodali komponente za kontekstualizacijo (situacijsko zavedanje) in umetno socialno inteligenco.

3.2.1 Pogovorni sistem SECA

Odprt pogovorni sistem SECA [20] je osnovan na programskem jeziku Python, njegova arhitektura ustreza zgoraj navedenim zahtevam za sistemske rešitve.

Oseba pristopi do komunikatorja in začne pogovor. Program v besedilo pretvorjen govor izpiše na zaslonu. Uporabnik potrди pravilnost izpisanega, ki se nato posreduje inteligenci pogovornega sistema (DialogFlow).

Program bo nato poskušal iz pridobljenega besedila prepoznati uporabnikov namen. Če v govoru osebe prepozna lokacijo, se to posreduje delu pridobivanja

podatkov. V primeru neuspešne prepoznavе komunikator prosi osebo, da stavek ponovi.

V delu pridobivanja podatkov je treba pridobiti informacije o položaju avtobusov in času potovanja avtobusa na željeno lokacijo, o trenutnem vremenu in temperaturi kraja, kjer je oseba prisotna, ter o času, ki bi ga oseba porabila, če bi na kraj lokacije hodila.

Pred odločanjem sistema, ali naj uporabnik uporabi javni prevoz ali ne, je treba preveriti verodostojnost prejetih podatkov. Potrди se prisotnost vseh potrebnih podatkov, v nasprotnem primeru se uporabniku sporoči o trenutni težavi njihovega pridobivanja.

Pri delu odločanja se glede na vrednosti podatkov program odloči, kako bo osebi svetoval. Če s senzorjem za hitenje zazna, da se osebi mudi in je javni prevoz hitrejši, komunikator svetuje, da oseba uporabi javni prevoz.

Posamezne komponente sistema so v shemo komunikatorja vključene tako, da se od programerja ne zahteva poznavanja celotne strukture, če želi zamenjati le eno komponento. Možno je tudi dodajanje nove funkcije, na primer za integracijo senzorja Kinect. Tako se olajša izvedba testiranja na daljavo.

Potek izvajanja programa se beleži v sledilno datoteko. To omogoča vpogled v uporabo komunikatorja, kamor so vključeni tudi neuspešni pogovori.

3.2.2 Jezikovne tehnologije: razpoznavanje in sinteza govora

Spletne storitve govornih tehnologij ponujajo veliko ponudnikov, vendar večina ne podpira slovenskega jezika. To je ena od glavnih ovir pri razvoju pogovornih sistemov s podporo v slovenskem jeziku.

Razpoznavanje govora ali pretvorba govora v besedilo (angl. Speech-to-text) je zmožnost stroja ali programa, da prepozna besede, izgovorjene na glas, in jih pretvori v

berljivo besedilo s pomočjo algoritmov. Osnovna programska oprema za prepoznavanje govora ima omejen besedni zaklad in lahko prepozna besede in besedne zveze le, če so jasno izgovorjene. Programska oprema za prepoznavanje govora se mora prilagoditi zelo spremenljivi in kontekstualno specifični naravi človeškega govora. Programski algoritmi, ki obdelujejo in organizirajo zvok v besedilo, se urijo na različnih govornih vzorcih, govornih slogih, jezikih, narečjih, naglasih in izrazih. Programska oprema prav tako loči govorjeni zvok od hrupa v ozadju, ki pogosto spremlja signal.

Sinteza govora (angl. Text-to-Speech) je umetna simulacija človeškega govora z računalnikom ali drugo napravo. Protipostavka razpoznavanja govora, sinteza govora, se večinoma uporablja za prevajanje besedilnih informacij v zvočne informacije in v aplikacijah, kot so glasovne storitve in mobilne aplikacije. Poleg tega se uporablja tudi v podporni tehnologiji za pomoč slabovidnim posameznikom pri branju besedilnih vsebin.

3.2.3 Relevantni kontekst pogovora

Relevantni kontekst so vsi podatki (v našem primeru čas, kraj in aktivnost), ki vplivajo na uporabnikove cilje in odločitve. Za demonstracijski dvogovor svetovanja mobilnosti v Ljubljani so bili to mesto in čas uporabnika, trenutni položaj avtobusov prevoznika LPP ter vremenska napoved.

Osnovni podatek, ki smo ga potrebovali za nadaljnje pridobivanje informacij, je bila trenutna lokacija uporabnika. Ker se je naša aplikacija komunikatorja navezovala na fizično napravo, ki je bila postavljena v javnem prostoru in ni spreminjala lokacije, smo podatek o začetni lokaciji vnesli že predhodno.

Za pridobivanje informacij o vremenu smo uporabili vmesnik uporabniškega programa (API) Open-Weather, dostopen na <https://openweathermap.org/>. API nam je podal vse kvantitativne podatke o vremenu, kot sta temperatura in vlaga, podal pa nam je tudi opisne podatke o oblačnosti in padavinah. Pridobljeni podatki so se nanašali na vnaprej vneseno lokacijo fakultete za elektrotehniko, kjer je bila naprava.

Za pridobivanje podatkov o poti in vseh podatkih o transportu smo uporabili vmesnik do Google zemljevidov. Maps (API). Za delovanje je bilo treba vnesti začetno in ciljno lokacijo. API nam je vrnil element JSON z vsemi relevantnimi podatki.

Za določitev optimalnega načina transporta (peš ali z javnim prevozom) smo primerjali trajanje poti. Velik vpliv na odločitev je imelo tudi vreme. Če je bilo vreme neprimerno (premrzlo ali so bile prisotne padavine ali veter), je bila pot z javnim prevozom izbrana ne glede na trajanje.

Pri uporabi Google Maps API je glavne težave povzročala oblika, v kateri so bili podatki pridobljeni, saj se je ta spreminjala glede na dostopnost avtobusov. Poleg tega je na obliko vplival tudi tip avtobusa (medkrajevni

ali mestni). Zaradi vseh možnih poti je bilo pravilno obliko težko predvideti.

3.2.4 Inteligenca pogovornega sistema

Za razvoj inteligence pogovornih sistemov obstaja večje število ponudnikov, ki ponujajo podobne rešitve. Razlikujejo se po tem, koliko jih lahko prilagajamo. Med njimi je Google DialogFlow, platforma za razumevanje naravnega jezika, ki olajša oblikovanje in integracijo pogovornega uporabniškega vmesnika v mobilno aplikacijo ali spletno aplikacijo; glejte <https://cloud.google.com/dialogflow/docs>. Na podlagi dobre dokumentacije za različne programske jezike hitro ustvarimo želeni potek dvogovora. Razvijalcem se trenutno ponuja 33 vnaprej zgrajenih agentov. Te predizdelane agente je mogoče po želji dodatno prilagoditi in usmeriti za določen primer uporabe. Ker imajo ti agenti že veliko nastavljenih namenov, to prihrani čas pri razvoju pogovornega sistema.

Poleg Googlove rešitve so v hitrem razvoju tudi druge rešitve. Gre za programske aplikacije, ki simulirajo človeški pogovor z glasovnimi ukazi ali besedilnimi klepeti ali oboje, ki ga je mogoče vgraditi in uporabiti s katerokoli večjo aplikacijo za sporočanje. Dva seznama takih aplikacij sta dostopna na povezavah <https://en.wikipedia.org/wiki/Chatbot> in <https://www.investopedia.com/terms/c/chatbot.asp>.

3.2.5 Umetna socialna inteligenca

Umetno socialno inteligenco predlaganega pogovornega sistema predstavljajo modeli strojnega učenja, ki senzorske signale uporabnika preslikajo v oceno izbranega socialnega signala uporabnika v realnem času, kot je hitenje. Hitenje namreč v osnovi določa, ali je za uporabnika pomembnejši čim krajši čas potovanja ali so relevantnejši drugi vidiki, kot je prijeten sprehod v lepem vremenu. Osnovna senzorja za določanje socialnega signala hitenja sta bila Kinect 3 [21] za določanje skeleta uporabnika in mikrofona za zajem govora. V zadnjem času se pojavljajo dovolj zmogljive kamere, ki z lastno strojno opremo dovolj hitro zaznavajo skelet uporabnika, npr.

<https://docs.luxonis.com/projects/hardware/en/latest/pages/BW1098OAK.html>.

V sodelovanju s študenti psihologije smo načrtali in izvedli eksperiment, pri katerem smo izdelali označeno učno množico za razvrščanje gibanja uporabnika v razreda »običajno gibanje« in »hitenje« na podlagi zajema skeleta uporabnika v realnem času. Z uporabo osnovne verzije klasifikatorja z metodo podpornih vektorjev smo dosegli dobre rezultate razvrščanja. Ovrednotenje tega razvrščanja v realnem okolju še ni na voljo.

3.2.6 Lokalni senzorji in terminal

Za komunikacijo z zunanjim terminalom smo izbrali arhitekturo spletnega strežnika z implementacijo Flask, glejte <https://flask.palletsprojects.com/>. Gre za vrsto spletnega ogrodja, ki je namenjeno temu, da se programerju ni treba ukvarjati z majhnimi podrobnostmi, poleg tega mu omogoča dostop do številnih knjižnic, ki jih bo potreboval pri delu. V velikem izboru spletnih ogrodij smo se odločili za Flask, ker je optimiziran za programski jezik Python, poleg tega pa je enostaven za uporabo.

Sestavljen je iz spletnega strežnika, gatewaya in vmesnika knjižnice Werkzeug, ki nam omogoči uporabo orodij za vzpostavitev povezave in komunikacije med strežnikom in odjemalcem, ter razširljivega mehanizma za predloge Jinja2 <https://jinja.palletsprojects.com/en/3.1.x/>. Jinja2 nam ponudi nabor predlog za izdelavo spletne strani in olajša dodajanje podatkovnih virov.

Ker je pri uspešnem pogovoru potrebna tudi neverbalna komunikacija, smo z uporabniškim vmesnikom prikazovali različne emotikone, ki so izražali čustveno stanje komunikatorja. Poleg tega je komunikator tudi sam zaznal, kdaj je oseba prenehala govoriti in začela čakati na odgovor.

3.2.7 Virtualizacija s sistemom Docker

Da bi zagotovili modularno izvedbo sistema, v primeru zahtevnejše implementacije pa tudi njegovo prožnost in odzivnost, smo arhitekturo sistema zasnovali na osnovi mikrostoritev. To pomeni, da delovanje aplikacije temelji na zbirki šibko povezanih granularnih storitev, ki med seboj podatke izmenjujejo prek lahkih komunikacijskih protokolov, kakršen je WebSocket.

Za razvoj in implementacijo mikrostoritev želimo okolje, v katerem lahko sodelujoče skupine programske module razvijajo neodvisno od lokalne konfiguracije razvojnega okolja, različic uporabljenih knjižnic in operacijskega sistema. Modularne gradnike spletnih storitev, od katerih vsak opravlja svoj del naloge, lahko prek spletnih ali datotečnih povezav nato integriramo v enovit komunikacijski scenarij. Take gradnike implementiramo kot storitve v gostiteljskem OS, pri čemer lahko pri množični uvedbi na različnih platformah pride do težav s prenosljivostjo, namestitvami, razširljivostjo in varnostjo. Z implementacijo posameznih servisnih modulov v ločene primerke navidezni računalnikov (VM) bi sicer lažje zagotovili prenosljivost in varnost, vendar ob visoki porabi strojnih virov.

Primerno rešitev ponuja sistem Docker [22], orodje, ki omogoča razvijalcem, sistemskim skrbnikom in drugim, da svoje aplikacije nameščajo v t. i. peskovnikih – v tem kontekstu jih imenujemo vsebniki (containers) –, ki se neodvisno in varno izvajajo znotraj gostiteljskega operacijskega sistema. Ključna prednost Dockerja je, da

lahko uporabniki svoje aplikacije skupaj z vsemi njenimi odvisnostmi zapakirajo v standardizirano enoto za razvoj programske opreme. V nasprotju z virtualnimi računalniki posamezni vsebniki ne nosijo bremena celotnega operacijskega sistema, in tako na gospodaren način izrabljajo vire gostiteljskega sistema.

V Dockerju namesto navidezni računalnikov osnovno entiteto predstavljajo vsebniki; slednji obratujejo na osnovi virov gostiteljskega operacijskega sistema, in tako prednosti izoliranega delovanja zagotavljajo ob le delčku porabe strojnih zmogljivosti. Vsebniki ponujajo logični mehanizem pakiranja, v katerem je mogoče aplikacije izolirati od okolja, kjer se dejansko izvajajo. Ta ločitev omogoča enostavno in konsistentno nameščanje aplikacij v vsebnikih ne glede na to, ali je ciljno okolje zasebni podatkovni center, javni oblak ali osebni prenosni računalnik razvijalca. Tako razvijalci delujejo v predvidljivem okolju, kjer so posamezne aplikacije od preostalih izolirane in jih je poleg tega mogoče izvajati kjerkoli. Vsebnike Docker podpirajo veliki ponudniki računalništva v oblaku (AWS, Azure in drugi).

Celoten koncept Dockerja ustreza modularnosti predlagane arhitekture komunikatorja. Na podlagi funkcionalne specifikacije posamezne mikrostoritve smo izdelali sliko Dockerja: to je datoteka samo za branje, ki vsebuje izvorno kodo, knjižnice, odvisnosti, orodja in druge datoteke, potrebne za delovanje aplikacije, in je sestavljena iz več plasti. Slika Docker je predloga za distribucijo storitve, na podlagi katere se ob namestitvi in zagonu ustvari in izvede t. i. vsebnik. Z dodajanjem zadnje bralno-pisalne plasti predhodnim slojem iz slike nastane vsebnik Docker (Docker Container), ki je entiteta virtualiziranega okolja izvajanja storitve in v katerem aplikacija teče izolirano od operacijskega sistema gostitelja.

4 RAZVOJ IN UPORABA PREDLAGANEGA KOMUNIKATORJA

V tem poglavju predstavljamo razvojni cikel komunikatorja in enostavna navodila za njegovo uporabo kot študijsko orodje.

4.1 Razvojni cikel komunikatorja

Po vzpostavitvi osnovne funkcionalnosti komunikatorja je razvojni cikel sestavljen iz korakov, predstavljenih v naslednjih poglavjih.

4.1.1 Postavitev in preverjanje scenarija uporabe

Gre za področje dvogovora, ki ga komunikator podpira, vključno s pokritimi scenariji uporabe, inteligenco dvogovora ter podprto umetno socialno inteligenco. Demonstracijski primer pokriva mobilnost v Ljubljani. Izvedljivost posameznega scenarija je odvisna od funkcionalnosti modula za določanje konteksta in funkcionalnosti modula za določanje socialnih signalov uporabnika v realnem času.

4.1.2 *Uporaba komunikatorja z zbiranjem testnih podatkov*

Uporaba komunikatorja z realnimi uporabniki nam omogoča zbiranje podatkov o uporabi, vključno z neuspešnimi dvogovori. S tem nastaja namenska učna množica: 1. za merjenje uspešnosti dvogovorov in 2. kot osnova za izdelavo učne množice strojnega učenja za izboljšavo modelov. Ta korak lahko vsebuje tudi eksplicitne uporabnikove odzive, kot je ocena kakovosti dvogovora glede na že izdelano metriko.

Jasno je, da zbiranje realnih podatkov zahteva že delujoči komunikator. Prvo različico komunikatorja zgradimo s pomočjo vgrajenih pravil inteligence komunikatorja in osnovno različico modulov za kontekstualizacijo ter za določanje socialnih signalov uporabnikov.

4.1.3 *Analiza delovanja in izboljšava*

Na podlagi podatkov, pridobljenih v prejšnjem koraku, s kombiniranjem strojne in ročne analize podatkov identificiramo scenarije, v katerih se komunikator ni ustrezno odzval. Glede na pogostnost in resnost napake izberemo ciljne scenarije in izboljšujemo delovanje posameznih modulov (kontekstualizacija, socialni signali, inteligenca pogovora). Glede na kompleksnost scenarijev in narave človeškega jezika je jasno, da popolno pokritje vseh scenarijev ni mogoče.

4.1.4 *Priprava izboljšanih modelov inteligence dvogovora in umetne socialne inteligence*

Glede na podatke prejšnje točke pripravimo novo različico učnih množic strojnega učenja (kontekstualizacija in razpoznavna socialnih signalov) ter z njimi pripravimo izboljšane modele strojnega učenja.

4.1.5 *Izboljšana različica komunikatorja*

Po osnovnih modulskih testiranjih novih modelov strojnega učenja te objavimo na virtualizaciji Docker in nova različica komunikatorja je nared za delovanje. Ponovni zagon sistema je podprt v sistemu Docker.

4.2 *Navodila za razvoj*

V tem podpoglavju predstavljamo tipične korake uporabe, uporabne tudi v okviru študijskega procesa na univerzitetnem nivoju.

4.2.1 *Vzpostavitev lokalnega razvojnega okolja*

Lokalno razvojno okolje vključuje razvojno okolje za programski jezik Python verzije 3.7.9 z razvojnimi orodji Visual Studio Code, GitHub in Docker. V pogovornem sistemu so integrirane govorne tehnologije, tehnologije situacijskega zavedanja (Google Maps API, OpenWeather API) in umetna socialna inteligenca (lastni model hitenja, USRP Radar, Kinect Senzor)

4.2.2 *Razvoj in modulska testiranja*

Razvoj in prva testiranja delovanja modulov ter programa kot celote potekajo v lokalnem okolju (na

osebnem računalniku). V zasnovi se deli pogovornega sistema (govorne tehnologije, situacijsko zavedanje in odločanje) razvijajo posamezno. Zaradi možnosti zamenjave oziroma nadgradnje dela kode v prihodnosti so deli v sistem vključeni tako, da se ne zahteva predznanja o strukturi, če bi se želelo delček programa zamenjati.

4.2.3 *Objava na virtualizaciji Docker*

Lokalno verzijo kode se po uspešnem testiranju objavi na virtualizaciji Docker. Namen objave je izvedba testiranja na daljavo. Zaradi konceptualnih ovir pri implementaciji celovite funkcionalnosti komunikatorja sta dekompozicija sistema na mikrostoritve in javna objava virtualizacije Docker še v razvoju.

4.2.4 *Testiranje sistema na daljavo*

Za optimizacijo testiranja pogovornega sistema se uporabi testne scenarije možnih interakcij osebe s pogovornim sistemom. Potek delovanja programa se skupaj z nastalimi napakami beleži v log datoteki. Po končanem testiranju se preveri dogodke, ko program ni pravilno deloval – neuspešna prepoznavna ulice, nezmožnost pridobivanja podatkov situacijskega zavedanja in napačnost odločanja – ter poskuša program izboljšati.

4.2.5 *Testiranje na fizičnem komunikatorju*

Testiranju sistema na daljavo sledi testiranje na fizičnem komunikatorju. Na tej točki se tudi implementira delovanje situacijskega zavedanja s pomočjo senzorjev – mikrofona, senzor Kinect.

Testiranje poteka v avli fakultete. Večji izziv te faze bi bil šum okolice, saj se ga težje loči od govora osebe.

5 ZAKLJUČEK IN NADALJNJE DELO

V članku predstavljamo pogovorni sistem, ki ga imenujemo socialni inteligentni komunikator. Gre za razvojno in demo okolje za umetno socialno inteligenco ter hkrati študijsko okolje na področju optimizacije in prilagajanja storitev uporabnikom.

Delujoč prototip na osnovi virtualizacije Docker [22] omogoča testiranje novih scenarijev in podpornih tehnologij ter izvajanje zahtevnejših študentskih projektov na zadevnem področju. Gradnja novih modelov umetne socialne inteligence je med drugim pomemben izziv na področju tehnološke oskrbe starejših in področju tehnološke podpore učenja in poučevanja.

Nadaljnje delo zajema razvoj novih modelov umetne socialne inteligence, vzpostavitev učinkovitejšega razvojnega okolja ter postavitev dveh demo aplikacij s področja oskrbe starejših [23] in merjenja multimedijske izpostavljenosti. Obenem načrtujemo uporabo sistema v študijskem procesu pri več predmetih v okviru študija na Fakulteti za elektrotehniko Univerze v Ljubljani, na študijskih programih Elektrotehnika in Multimedija.

LITERATURA

- [1] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A Survey on Neural Speech Synthesis," 2021, [Online]. Available: <http://arxiv.org/abs/2106.15561>.
- [2] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, "Automatic speech recognition: a survey," *Multimed. Tools Appl.*, vol. 80, no. 6, pp. 9411–9457, 2021, doi: 10.1007/s11042-020-10073-7.
- [3] M. Lohani, B. R. Payne, and D. L. Strayer, "A review of psychophysiological measures to assess cognitive states in real-world driving," *Front. Hum. Neurosci.*, vol. 13, no. March, pp. 1–27, 2019, doi: 10.3389/fnhum.2019.00057.
- [4] J. A. Castro-García, A. J. Molina-Cantero, I. M. Gómez-González, S. Lafuente-Arroyo, and M. Merino-Monge, "Towards Human Stress and Activity Recognition: A Review and a First Approach Based on Low-Cost Wearables," *Electron.*, vol. 11, no. 1, pp. 1–30, 2022, doi: 10.3390/electronics11010155.
- [5] D. Jurafsky, "Conversational Agents AKA Dialog Agents," 2018.
- [6] Google, "Google Assistant." 2022, [Online]. Available: <https://assistant.google.com/>.
- [7] Amazon, "Amazon Alexa." 2022, [Online]. Available: <https://alexa.amazon.com>.
- [8] Microsoft, "Microsoft Cortana." 2022, [Online]. Available: <https://www.microsoft.com/en-us/cortana>.
- [9] Apple, "Apple Siri." 2022, [Online]. Available: <https://www.apple.com/siri/>.
- [10] J. Dötterl, R. Bruns, and J. Dunkel, "Incorporating situation awareness into recommender systems," in *ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems*, 2017, vol. 2, pp. 676–683, doi: 10.5220/0006385106760683.
- [11] E. A. Oguz and A. Košir, "Multimedia-content-index based experimental content selection," in *Proceedings of the 15th International Symposium on Operations Research*, SOR 2019, 2019.
- [12] W. Sheikh and A. Ghafoor, "Jitter-Minimized Reliability-Maximized Joint Optimization Quality-of-Service Routing for GPS Networks," *ECE Tech. Reports*, p. 348, 2007, Accessed: 24-Jun-2011. [Online]. Available: <http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1348&context=ecetr>.
- [13] A. Vinciarelli, M. Pantic, and H. Bourlard, "Social signal processing: Survey of an emerging domain," *Image Vis. Comput.*, vol. 27, no. 12, pp. 1743–1759, Nov. 2009, doi: 10.1016/j.imavis.2008.11.007.
- [14] A. Vinciarelli et al., "Bridging the gap between social animal and unsocial machine: A survey of social signal processing," *IEEE Trans. Affect. Comput.*, vol. 3, no. 1, pp. 69–87, 2012, Accessed: 23-Apr-2012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5989788.
- [15] P. van der Wel and H. van Steenbergen, "Pupil dilation as an index of effort in cognitive control tasks: A review," *Psychon. Bull. Rev.*, vol. 25, no. 6, pp. 2005–2015, 2018, doi: 10.3758/s13423-018-1432-y.
- [16] E. Butkvečičiute, M. Erinš, and L. Bikulčiene, "An adaptable human fatigue evaluation system," in *Procedia Computer Science*, 2021, vol. 192, pp. 1274–1284, doi: 10.1016/j.procs.2021.08.131.
- [17] H. Gunes, B. Schuller, M. Pantic, and R. Cowie, "Emotion representation, analysis and synthesis in continuous space: A survey," 2011 *IEEE Int. Conf. Autom. Face Gesture Recognit. Work. FG* 2011, pp. 827–834, 2011, doi: 10.1109/FG.2011.5771357.
- [18] N. Jia and C. Zheng, "Emotion Speech Synthesis Method Based on Multi-Channel Time-Frequency Domain Generative Adversarial Networks (MC-TFD GANs) and Mixup," *Arab. J. Sci. Eng.*, vol. 47, no. 2, pp. 1749–1762, 2022, doi: 10.1007/s13369-021-06090-9.
- [19] T. Vodlan, M. Tkalčič, and A. Košir, "The impact of hesitation, a social signal, on a user's quality of experience in multimedia content retrieval," *Multimed. Tools Appl.*, vol. 74, no. 17, 2015, doi: 10.1007/s11042-014-1933-2.
- [20] D. Tellols, M. Lopez-Sanchez, I. Rodríguez, P. Almajano, and A. Puig, "Enhancing sentient embodied conversational agents with machine learning," *Pattern Recognit. Lett.*, vol. 129, pp. 317–323, 2020, doi: 10.1016/j.patrec.2019.11.035.
- [21] Microsoft, "Microsoft Azure Kinect Sensor," 2022, [Online]. Available: <https://azure.microsoft.com/en-us/services/kinect-dk/>.
- [22] Docker Inc., "Virtualisation Docker," 2022, [Online]. Available: Docker Inc.
- [23] A. Košir, "Sodobne tehnologije in reševanje staranja prebivalstva," *Kakov. STAROST*, vol. 4, no. 22, 2019.

Emma Masterl je študentka magistrskega programa Biomedicinske tehnike na Fakulteti za elektrotehniko Univerze v Ljubljani. Trenutno raziskovalno delo opravlja v Laboratoriju za slikovne tehnologije. Njena interesna področja obsegajo obdelavo medicinskih slik in globoko učenje.

Samo Jean je študent magistrskega programa IKT na Fakulteti za elektrotehniko Univerze v Ljubljani. Sodeluje pri projektih v Laboratoriju za uporabniku prilagojeno komunikacijo in ambientno inteligenco. Njegova interesna področja zajemajo spletni razvoj in strojno učenje.

Jure Potočnik je študent magistrskega programa IKT na Fakulteti za elektrotehniko Univerze v Ljubljani. Njegova interesna področja med drugim zajemajo mobilna omrežja ter avtomatizacijo procesov.

Urban Burnik je docent na Fakulteti za elektrotehniko Univerze v Ljubljani. Predava dodiplomske predmete s področij IKT, digitalne obdelave signalov, mobilnih komunikacij in multimedije. Njegove raziskovalne dejavnosti vključujejo večdimenzionalno obdelavo signalov, telekomunikacijske storitve, personalizirane multimedijske storitve in interaktivno televizijo.

Andrej Košir je redni profesor na Fakulteti za elektrotehniko Univerze v Ljubljani in vodja Laboratorija za uporabniku prilagojeno komunikacijo in ambientno inteligenco. Njegova področja dela obsegajo optimizacijo, prilagajanje uporabnikom, merjenje in obdelavo socialnih signalov uporabnikov, tehnologije za podporo učenju in tehnologije za podporo starejšim.