

UNIVERZITET U BEOGRADU  
MATEMATIČKI FAKULTET



Dara Milojković

**INTELLIGENTNI DIGITALNI SAGOVORNIK  
ZASNOVAN NA SEKVENCAMA SA  
MEHANIZMOM PAŽNJE**

master rad

Beograd, 2021.

**Mentor:**

dr Jelena GRAOVAC, docent  
Univerzitet u Beogradu, Matematički fakultet

**Članovi komisije:**

dr Jovana KOVAČEVIĆ, docent  
Univerzitet u Beogradu, Matematički fakultet

dr Aleksandar KARTELJ, docent  
Univerzitet u Beogradu, Matematički fakultet

**Datum odbrane:** \_\_\_\_\_

*Zahvaljujem se profesorki Jeleni Graovac  
za pruženu podršku i istrajnost,  
kolegama Anđelki Zečević, Anđelki Milovanović i Neveni  
na pomoći i bodrenju, Ivanu na savetima,  
a porodici na strpljenju*

**Naslov master rada:** Inteligentni digitalni sagovornik zasnovan na sekvencama sa mehanizmom pažnje

**Rezime:**

Digitalni sagovornici poslednjih godina privlače veliku pažnju kako kompanija tako i korisnika usluga. Od sredine dvadesetog veka pa do danas našli su mesto u svakoj sferi poslovanja, naročito u oblastima koje su zasnovane na stalnoj komunikaciji sa korisnicima. Razvojem tehnologija, mnoge uloge koje su nekada pripadale isključivo ljudima preuzimaju digitalni sagovornici. Upotrebom digitalnih sagovornika mogu se rešiti problemi obrade lakih zahteva korisnika čime se rasterećuju osnovni tokovi tehnološke podrške i omogućava brza usluga na opšte zadovoljstvo kompanije i korisnika usluge. Sa druge strane, sve veću popularnost ostvaruju digitalni sagovornici otvorenog domena, odnosno, digitalni sagovornici koji nisu vezani za jednu oblast poslovanja ili temu razgovora. Ovakvi digitalni sagovornici su namenjeni za pružanje emotivnih i socijalnih usluga korisnicima. Ipak, iako su veoma korisni, digitalni sagovornici ne mogu da zamene čoveka i njihova uloga se uglavnom vezuje za neke lake operacije kao što su preporuke proizvoda i tehnička podrška za rešavanje poznatih problema, dok su složenije usluge i dalje rezervisane za ljude.

Postoje mnogi kriterijumi podele digitalnih sagovornika, među kojima je i podela na inteligentne ili zasnovane na pravilima. Digitalni sagovornici vođeni pravilima su vezani za bazu u kojoj se nalaze predefinisani odgovori na svako moguće pitanje. To znači da prilikom kreiranja ove baze treba da se predvidi svaki mogući ishod razgovora, što je u mnogim slučajevima nemoguće. Zbog toga je često prilikom komunikacije sa ovakvom vrstom digitalnih sagovornika neophodna ljudska pomoć, što može da izazove nelagodnosti kod korisnika i kod kompanija. Za razliku od njih, inteligentni digitalni sagovornici su zasnovani na nekoj metodi mašinskog učenja. Ovakva vrsta sagovornika uči iz komunikacije sa korisnicima i stalno napreduje. Međutim, njihova implementacija nije jednostavna, pre svega jer je potrebna integracija senzibilnosti koje ljudska bića prirodno imaju i implementacija mogućnosti razlikovanja raznih semantički sličnih upita i modernih izraza.

Sve popularnija metoda mašinskog učenja koja se koristi za implementaciju inteligentnih digitalnih sagovornika su neuronske mreže. Neuronske mreže predstavljaju sistem sastavljen od jednostavnih jedinica koje međusobno mogu da razmenjuju podatke stvarajući pritom kompleksnu vezu tih podataka. Njihova snaga leži u sposobnosti dobrog prilagođavanja ulaznim podacima, učenja i generisanja odgovora sa

velikom tačnošću. U ovom radu biće predstavljena implementacija neuronske mreže zasnovane na sekvencama sa mehanizmom pažnje koja će biti obučena i testirana na skupu dijaloga sa Univerziteta Kornel. Arhitektura koja se koristi za dizajniranje mreže je enkoder–dekoder arhitektura u kojoj glavne komponente čine dve neuronske mreže. Kako bi se omogućio rad sa dužim rečenicama koristi se Bahdanaov mehanizam pažnje. Rezultat obučavanja biće digitalni sagovornik koji će komunicirati sa korisnicima na engleskom jeziku.

**Ključne reči:** digitalni sagovornik, neuronska mreža, LSTM, Bahdanaov mehanizam pažnje

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Pregled razvoja digitalnih sagovornika</b>	<b>4</b>
<b>3</b>	<b>Tehnike pravljenja digitalnih sagovornika</b>	<b>7</b>
3.1	Sagovornici vođeni pravilima . . . . .	7
3.2	Inteligentni sagovornici . . . . .	11
<b>4</b>	<b>Neuronske mreže</b>	<b>14</b>
<b>5</b>	<b>Implementacija</b>	<b>25</b>
5.1	Preprocesiranje podataka . . . . .	25
5.2	Podešavanje parametara neuronske mreže . . . . .	30
5.3	Obučavanje i evaluacija modela . . . . .	33
<b>6</b>	<b>Eksperimentalni rezultati</b>	<b>35</b>
<b>7</b>	<b>Zaključak</b>	<b>39</b>
	<b>Bibliografija</b>	<b>41</b>

# Glava 1

## Uvod

Godine 1947. Alan Tjuring je napisao rad u kome je postavio veoma zanimljivo pitanje koje je uzdrimalo tadašnju javnost: „Mogu li mašine da misle?“. Čuveni matematičar je u mašinama video veliki potencijal koji je, u vremenu kada su se mašine tretirale kao veliki električni abakusi, daleko prevazilazio tadašnje mogućnosti računanja i memorisanja informacija. Razmišljao je o mašini ili programu koji može da prevaziđe inicijalno zadata pravila i uči kroz iskustvo. Tjuring je dao predlog testa čiji je osnovni cilj bio ispitivanje sposobnosti neke mašine da demonstrira inteligenciju ekvivalentnu ili nerazlučivu od ljudske. Test se sastojao iz toga da se ljudski sudija upušta u konverzaciju, prirodnim jezikom, sa jednom mašinom i sa jednim čovekom, pri čemu oba sagovornika pokušavaju da se predstave kao ljudi. Ukoliko sudija nije u mogućnosti da sa sigurnošću odredi koji od dva sagovornika je čovek, a koji mašina, onda je mašina prošla test. Ovaj test je postao važan koncept u filozofiji veštačke inteligencije, oblasti kojoj pripada i razvoj digitalnih sagovornika.



Slika 1.1: Digitalni sagovornik <sup>1</sup>

U današnje vreme se često čuje fraza „vreme je novac”. Užurbani način života je, između ostalog, posledica razvoja tehnologija i mogućnosti masovne proizvodnje i potrošnje koji su sastavni deo naših života. Kako bi se korisnici prilagodili na stalne promene tehnologija, kompanije su zapošljavale sve veći broj ljudi koji su obavljali ulogu operatera. Nedovoljan broj radnika je često bio uzrok višerasovnom čekanju na razgovor i dobijanje adekvatne usluge, što je izazivalo nezadovoljstvo kod korisnika usluga. Kako bi ovakvo stanje prevazišli, kompanije su sve više počele da koriste automatizovanu alternativu radnicima – digitalne sagovornike.

Digitalni sagovornici (eng. Chatbot) su programi koji na osnovu ulaza koji predstavlja pitanje na prirodnom jeziku generišu izlaz, koji predstavlja odgovor, takođe napisan na prirodnom jeziku. Osnovna ideja programa je da se simulira ljudski način komunikacije i interakcije sa korisnicima bez pomoći programera. U poslednjih nekoliko godina digitalni sagovornici su našli mesto pružaoca usluga u raznim kompanijama, bilo da je reč o kupoprodajnim uslugama, tehničkoj podršci ili medicinskoj pomoći. Time kompanije ne troše vreme i novac na zapošljavanje ljudi, dok korisnici dobijaju brzu uslugu. Ovakvi sagovornici su obično specifični za neku oblast i mogu biti veoma efikasni. Mnoge aplikacije na pametnim uređajima su namenjene pružanju usluga emotivne i psihološke podrške korisnicima kojima je to potrebno i predstavljaju sagovornike bez domena (eng. Open-domain chatbot) [17]. Sagovornike bez domena je teže implementirati jer je potrebna velika baza podataka kako bi uključila što veći rečnik. Teži se implementaciji ponašanja koje je tipično za čoveka prilikom komunikacije: pravljenje dugačkih pauza, promena toka komunikacije ili promena emotivnog stanja korisnika, kao i slovne i gramatičke greške. I dalje se traga za boljim i naprednijim sagovornicima koji bi mogli ne samo da konstruišu rečenicu kao što bi čovek to uradio, već i da tok razgovora usmere tako da bude nalik na ljudski.

Postoji veliki broj tehnika za konstrukciju digitalnih sagovornika među kojima se nalaze i inteligentni digitalni sagovornici koji su zasnovani na upotrebi neuronskih mreža. U ovom radu biće prikazan model neuronske mreže koja će služiti kao osnova jednog inteligentnog digitalnog sagovornika otvorenog domena. Baza podataka koja će biti korišćena za obučavanje mreže je skup dijaloga sa Univerziteta Kornel u Njujorku koji sadrži kolekciju izmišljenih razgovora iz sirovih filmskih scenarija [5]. U narednoj glavi biće prikazan pregled razvoja digitalnih sagovornika kroz istoriju. U glavi 3 biće uvedena definicija inteligentnih digitalnih sagovornika,

---

<sup>1</sup>Slika je preuzeta sa <https://www.zfort.com/blog/top-10-best-ai-chatbots>.



kao i razlog odabira inteligentnog digitalnog sagovornika naspram ostalih vrsta sagovornika (konkretno digitalnih sagovornika vođenih pravilima). Za potrebe ovog rada koristiće se model neuronske mreže zasnovan na sekvencama sa Bahdanao-  
vim mehanizmom pažnje. Sama implementacija modela, struktura modela i obrada podataka će biti opisane u glavama 4 i 5. Nakon odabira parametara sledi obučavanje i rezultati obučavanja će biti prikazani u glavi 6. Na kraju, rad završavamo zaključkom.

## Glava 2

# Pregled razvoja digitalnih sagovornika

Reč „Eureka” potiče iz starogrčkog jezika i u slobodnom prevodu znači „Uspeo sam” ili „Izumeo sam”, a vezuje se za poznatog matematičara Arhimeda. Istorija je puna ovakvih „eureka” momenata. Najviše ih je bilo tokom prve industrijske revolucije koja je omogućila razvoj društva u kome danas živimo, kao i razvoj novih tehnologija bez kojih je savremeni svet nezamisliv.

Jedan takav momenat desio se i 1966. godine kada se pojavio prvi program koji je imao ulogu psihoterapeuta, ELIZA [16]. Jozef Vajcenbaum (*eng.* Joseph Weizenbaum) je bio profesor na MIT-u (*eng.* Massachusetts Institute of Technology) i dao je sebi zadatak da napravi program koji će da uveri čoveka da priča sa drugim čovekom. ELIZA je dizajniran da prepozna ključne reči u rečenici i na osnovu njih ispiše predefinisani odgovor. Na primer, ukoliko je čovek u svojoj rečenici napisao reč majka, program bi odgovorio „Pričajte mi o vašoj porodici”. Baza koja je korišćena za interakciju je sadržala samo termine koje su psihoterapeuti koristili u svojim sesijama. U tom periodu ovakva interakcija sa programom, pa i sa kompjuterom, je bila ispred svog vremena jer su se prvi korisnički računari pojavili osamdesetih godina. Eksperiment je doživeo veliki uspeh i cilj je bio postignut i to u tolikoj meri da su se neki ljudi koji su imali ulogu sudije čak emotivno vezali za program. Međutim, mana ovog programa je bila ograničenost teme za razgovor i šablonski odgovori iz male baze podataka. Naravno, baza podataka je mogla da se proširi ali bi i dalje bila limitirana na uvežbane odgovore.

Prošlo je dosta vremena dok se nije video pomak u razvoju digitalnih sagovornika. Posle programa ELIZA pojavio se i program PARRY, koji je dobio ulogu pacijenta

sa šizofrenijom kako bi se testirali psihoterapeuti [4]. Međutim, iako se mislilo da je razvijeniji od ELIZA programa zbog dodatnih emocija, nije prošao testiranje jer nije uspeo da ubedi testere da je ljudsko biće. Godine 1995. nastao je program ALICE (*eng.* Artificial Linguistic Internet Computer Entity) prvi sagovornik koji je radio na Internetu i koji je pisan na jeziku AIML (*eng.* Artificial Intelligence Markup Language) [1]. AIML je forma XML-a (*eng.* Extensible Markup Language) i specijalno je napisan za kreiranje virtualnih sagovornika [15]. Ovaj jezik je omogućio pravljenje šablona koji prepoznaju bitne reči u rečenici (*eng.* template) i šablona koji omogućavaju dopunjavanje odgovora (*eng.* pattern). Međutim, iako je imao veću bazu podataka i način odgovaranja je kompleksniji nego kod programa ELIZA, on je i dalje zavisio od čoveka i nije mogao da uči iz prethodnog iskustva.

Dvadeseti vek je svedok razvoja baza podataka i njihovih primena kako u kompanijama tako i u domovima ljudi. Novi milenijum je doveo nove inovacije u polju digitalnih sagovornika i te promene su se dešavale brzo. Sa nastankom Interneta potreba za bazama je postala veća, pa su se pojavile veće i korisnije baze za korisnike. Ovo naglo unapređenje baza se odrazilo i na razvoj digitalnih sagovornika. U prvoj dekadi dvadeset prvog veka nastali su programi koji su povezivali korisnika sa bazama podataka čime je digitalni sagovornik dobio mogućnost da sa korisnikom „razgovara” o različitim temama, počev od filmova i sportova do vesti, vremena i dešavanja na berzi. Neki digitalni sagovornici su programirani da prepoznaju ljudski glas i mimike lica kako bi sakupili što više informacija i tako pokušali da ostave utisak što boljeg razumevanja. Velike kompanije, kao što su Epl (*eng.* Apple), Gugl (*eng.* Google), Majkrosoft (*eng.* Microsoft), Amazon (*eng.* Amazon), su dosta utrošile na razvoj virtuelnih sagovornika odnosno virtuelnih pomagača, međutim, i pored toga, ti programi nisu idealni. Majkrosoft je dizajnirao Kortanu (*eng.* Cortana) koja se aktivira glasom i izvršava praktične zadatke na zahtev korisnika, međutim, jedna od njenih velikih mana je što je u mogućnosti da instalira maliciozni softver. Siri (*eng.* Siri) je digitalni sagovornik kompanije Epl koji je takođe napravljen da prepoznaje ljudski govor. Može da se adaptira korisnikovim željama i prethodnim pretragama kako bi pružio ugodnije korišćenje. Ali i ovaj digitalni sagovornik ima ograničenja, kao na primer, ne može da prepozna akcente i ne radi bez Internet konekcije.

Do nagle promene načina korišćenja digitalnih sagovornika došlo je početkom 2016. godine jer se javila potražnja povećane komunikacije korisnika sa proizvođačima i pružaocima različitih usluga. Na mnogim sajtovima su se pojavili asistenti koji su digitalni sagovornici i odgovaraju na pitanja koja postavljaju korisnici. Hi-

ljade sagovornika je napravljeno za potrebe marketinga, edukacije, zabave, industrije, zdravstvenog sistema i raznih drugih sistema za podršku. Njihove glavne prednosti su te što ne oduzimaju vreme zaposlenim ljudima u kompaniji i jeftiniji su. Moderni digitalni sagovornici i asistenti se prave tako da imaju mogućnost učenja iz prethodnih iskustava čime je napravljen veliki pomak u dizajniranju digitalnih sagovornika otvorenog domena kao što je Šaoce (eng. Xiaolce) koji je dizajniran da bude empatični socijalni digitalni sagovornik [17].

## Glava 3

# Tehnike pravljenja digitalnih sagovornika

Digitalni sagovornici imaju dosta primena u industriji i njihove tehnike dizajniranja, kao i veličina odgovarajuće baze koja će biti korišćena, zavise od tih primena. U skladu sa tim, digitalni sagovornici se dele na [11]:

**Sagovornike vođene pravilima** – koji komuniciraju sa korisnicima uz pomoć već pripremljene baze podataka koja često sadrži odgovore vezane za neku konkretnu temu razgovora. Veoma su jednostavni za implementaciju ali su ograničeni dimenzionalnošću.

**Inteligentne sagovornike** – koji komuniciraju sa korisnicima generisanjem svojih odgovora na osnovu prethodno naučenih primera. Za konstrukciju ove vrste digitalnih sagovornika često se koriste metode mašinskog učenja i za razliku od prethodno pomenutih digitalnih sagovornika vođenih pravilima, nisu ograničeni jednom temom razgovora.

Obe tehnike imaju svoje prednosti i mane, a kompanije se opredeljuju za onu tehniku koja više zadovoljava njihove potrebe.

### 3.1 Sagovornici vođeni pravilima

Sagovornici vođeni pravilima odgovaraju na pitanja koja su zasnovana na predefinisanim skupu definisanih pravila [11]. Pogodni su za obavljanje nekih veoma jednostavnih operacija kao što su rezervisanje mesta u hotelu, prijavljivanje grešaka

ili zakazivanje pregleda kod lekara. Prilikom razgovora sa korisnikom koriste se propratna pitanja kako bi se dobio što bolji uvid o pravom korisničkom zahtevu. Tok razgovora podseća na drvoliku strukturu gde su čvorovi pitanja digitalnog sagovornika, a grane odgovori korisnika (slika 3.1).



Slika 3.1: Primer razgovora sagovornika vođenih pravilima

Kompleksnost pravila zavisi od oblasti u kojoj će biti primenjeni kao i od veličine i strukture skupa. Skup podataka omogućava brzu i laku implementaciju digitalnih sagovornika kao i njihovu kontrolu.

Međutim, ono što čini ove sagovornike dobrim i uspešnim takođe može da ima i negativno dejstvo. Samim tim što su vođeni pravilima nisu sposobni da obrade one rečenice ili radnje korisnika koje se ne nalaze u skupu. Na primer, rečenice „Hteo bih da zakažem pregled kod doktora.” i „Zakazao bih pregled kod lekara.” su semantički iste ali ako se ne nalaze u bazi sagovornik neće znati adekvatan odgovor. Često je tok razgovora „hladan” i bez dodatnih interakcija što ponekad čini komunikaciju sa korisnikom neugodnom [12]. Među poznatijim sagovornicima vođenih pravilima nalaze se i sagovornici zasnovani na obrascima.

### Algoritmi zasnovani na obrascima

Među prvim tehnikama koje su se koristile za pravljenje virtuelnih sagovornika je algoritam zasnovan na obrascima (eng. pattern matching algorithm) [2]. Karakteristika ove tehnike je sama struktura baze podataka koja se ogledala u korišćenju obrazaca za pitanja (eng. patterns) i obrazaca za odgovore (eng. templates). Jezik koji se koristi za prepoznavanje ovakvih obrazaca je AIML jezik, koji je prvobitno korišćen za ALICE program. Osobina ovog jezika je hijerarhijska dokumentacija

pitanja koje korisnik može da postavi, i odgovora na ta pitanja, kao i njihova jednostavna pretraga prilikom komunikacije sa korisnikom. Hijerarhijska struktura je zasnovana na pravljenju etiketa koje predstavljaju naredbe jezika.

U primeru 3.1 data su dva načina struktuiranja obrazaca koja najbolje pokazuju njihovo uopšteno ponašanje bez ulaženja u preprocesiranje rečenica.

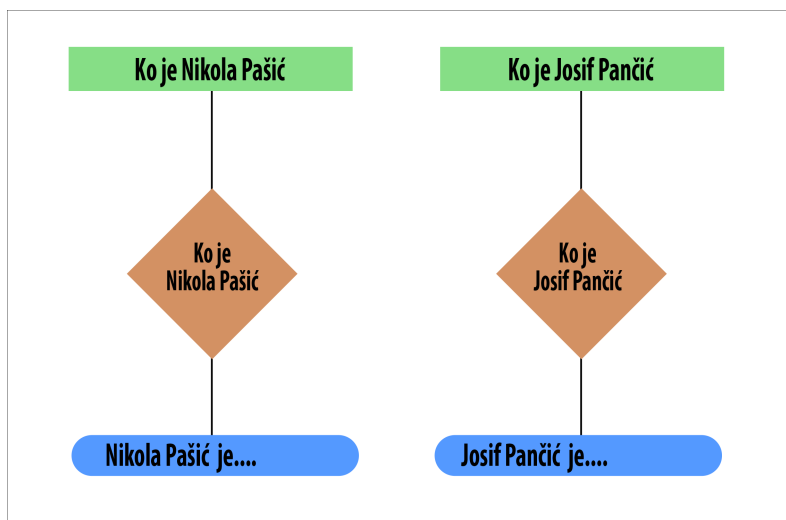
```
<aiml version="1.0.1" encoding="UTF-8">
  <category>
    <pattern>Ko je Josif Pančić</pattern>
    <template>
      Josif Pančić je bio srpski lekar, botaničar i prvi
      predsednik Srpske kraljevske akademije. Otkrio je
      novu vrstu četinara koja je po njemu nazvana Pančić
      eva omorika.
    </template>
  </category>

  <category>
    <pattern>Ko je Nikola Pašić</pattern>
    <template>
      Nikola Pašić je bio srpski i jugoslovenski političar i
      dugogodišnji predsednik vlade Kraljevine Srbije i
      Kraljevine Srba, Hrvata i Slovenaca.
    </template>
  </category>

  <category>
    <pattern>Da li znaš ko je * </pattern>
    <template>
      <srai> Ko je <star/> </srai>
    </template>
  </category>
</aiml>
```

#### Primer 3.1: AIML jezik

Ukoliko je pitanje koje je korisnik uneo „Ko je Nikola Pašić?“, baza se pretražuje sa ciljem da se pronađe pitanje u izvornom obliku. Ukoliko takvo pitanje postoji, koristi se obrazac za odgovor *<template>Nikola Pašić je...</template>* koji se nalazi ispod obrasca za pitanje *<pattern>Ko je Nikola Pašić</pattern>*. Dijagram toka pretrage može se videti na slici 3.2. U slučajevima kada je moguće različito izgovaranje ili pisanje jedne misli (kao što su, na primer: „zdravo“, „ćao“, „dobar dan“ različiti načini pozdravljanja) koristi se etiketa *<srai></srai>* u kombinaciji

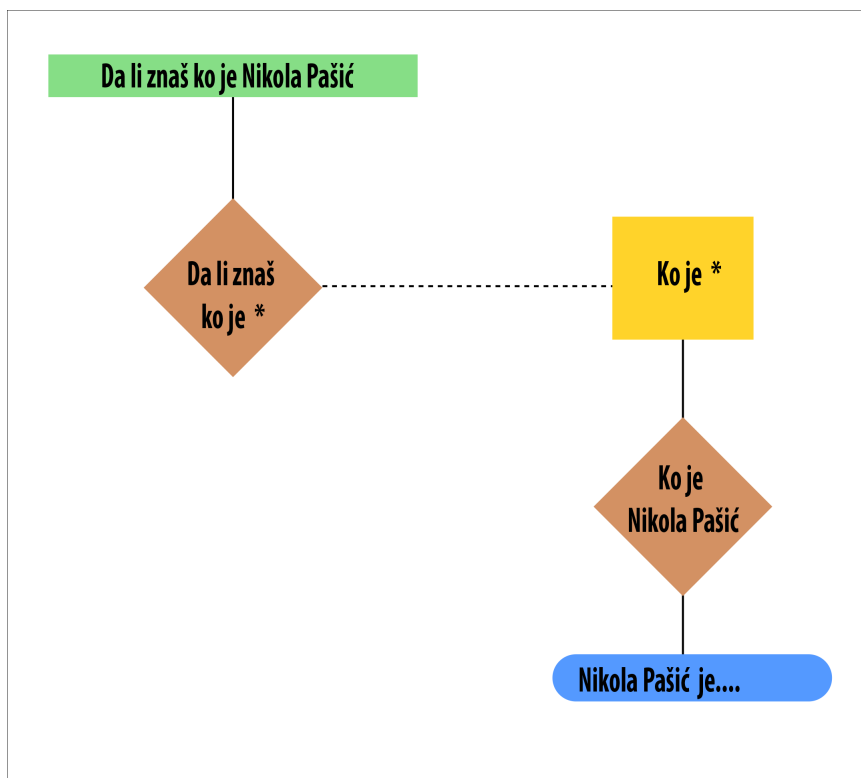


Slika 3.2: Primer dohvatanja odgovora na pitanje „Ko je Nikola Pašić”

sa oznakom `*` (oznaka `</star>`). Ukoliko pretpostavimo da je korisnik uneo pitanje u sastavu neke druge rečenice, na primer „Da li znaš ko je Nikola Pašić?”, suština rečenice ostaje ista ali je drugačije napisana. Ovakav slučaj se može rešiti na dva načina:

- Prvo rešenje problema je da se napravi novi obrazac za pitanje koji će imati sadržaj „Da li znaš ko je Nikola Pašić?”, ali to onda podrazumeva pravljenje obrazaca za svaki pojedinačan slučaj (ukoliko postoji obrazac „Ko je Josif Pančić” potrebno je napraviti obrazac i za pitanje „Da li znaš ko je Josif Pančić”).
- Drugi način je apstraktovanje rečenice „Da li znaš ko je ” dodavanjem specijalnog simbola `*` koji označava ostatak rečenice ili određenu sekvencu reči. U ovom primeru simbol se dodaje na kraj rečenice jer je pretpostavka da se tu nalazi nečije ime. Obrazac za odgovor na ovo pitanje sadrži etiketu `<srail></srail>` sa sadržajem `Ko je </star>`. Etiketom `</star>` dohvata se simbol `*` iz obrasca za pitanje, u ovom slučaju je to „Nikola Pašić”, te sadržaj etikete `<srail></srail>` postaje „Ko je Nikola Pašić” koji se potom pretražuje kao obrazac za pitanje. Ovim postupkom dohvatanja sadržaja pitanja i njihovim ponovnim pretraživanjem, ispoljava se rekurzivna strana ove strukture podataka koja omogućava laku pretragu i održavanje. Primer dijagrama toka pretrage se može videti na slici 3.3.





Slika 3.3: Primer dohvaćanja odgovora na pitanje „Da li znaš ko je Nikola Pašić“

Očigledne mane ovakvog pristupa su potreba za definisanjem velikog broja pojmova unutar same strukture, njeno održavanje, nemogućnost generisanja odgovora koji se ne nalazi u bazi i nefleksibilnost digitalnog sagovornika na promene korisničkih rečenica (na primer, slovne i gramatičke greške).

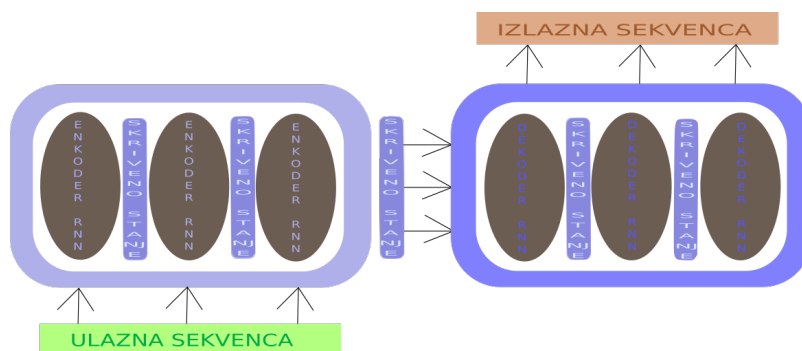
## 3.2 Inteligentni sagovornici

Inteligentni sagovornici, za razliku od sagovornika vođenih pravilima, ne koriste pomoćne putokaze prilikom konstruisanja odgovora. Ovakvi digitalni saradnici izvlače smisao korisničkih rečenica traženjem veza između reči u rečenici pri čemu se često koriste metode mašinskog učenja i veliki skup podataka (eng. big data) [2]. U najboljem slučaju sagovornik treba da razume gramatičke greške i greške nastale prilikom kucanja ili govora i da, bez obzira na grešku, ispravno odgovori bez pomoći programera. Nažalost, kako bi se napravio veliki skup podataka potrebno je dosta vremena kako za prikupljanje podataka tako i za treniranje. Uz naprednije tehnike, sagovornik može i da protumači emocije korisnika i da na osnovu njih prikladno

reaguje. Takve vrste sagovornika u korisničkim servisima postižu veliki uspeh i mogu da se snađu u nepredviđenim situacijama [17]. Kod inteligentnih sagovornika, potrebno je dosta vremena za obučavanje kako bi se dobio odgovarajući nivo razumevanja konteksta. Baza podataka je veoma bitna jer ukoliko prilikom obučavanja dobije neke neprikladne rečenice, naučice i da generiše slične neprikladne odgovore. Mnogi inteligentni sagovornici koriste napredne tehnike koje pripadaju oblasti obrade prirodnih jezika i modele zasnovane na neuronskim mrežama.

### Model zasnovan na sekvencama

Ovaj model kao ulaz prihvata rečenicu i generiše izlaz na osnovu sekvence reči koje nastaju raščlanjivanjem prihvaćene rečenice [13]. Za implementacija ovog modela koriste se neuronske mreže, odnosno rekurentne neuronske mreže. Model se sastoji iz dve zasebne neuronske mreže koje se zovu enkoder i dekoder (slika 3.4). Uloga enkodera je da izvrši analizu ulazne rečenice i nađe smisao rečenice. Rečenica se prebacuje u vektor brojeva i obrađuje se u skrivenim slojevima mreže stvarajući vektor skrivenog stanja. Taj vektor se prosleđuje dekoderu i fiksne je dužine.

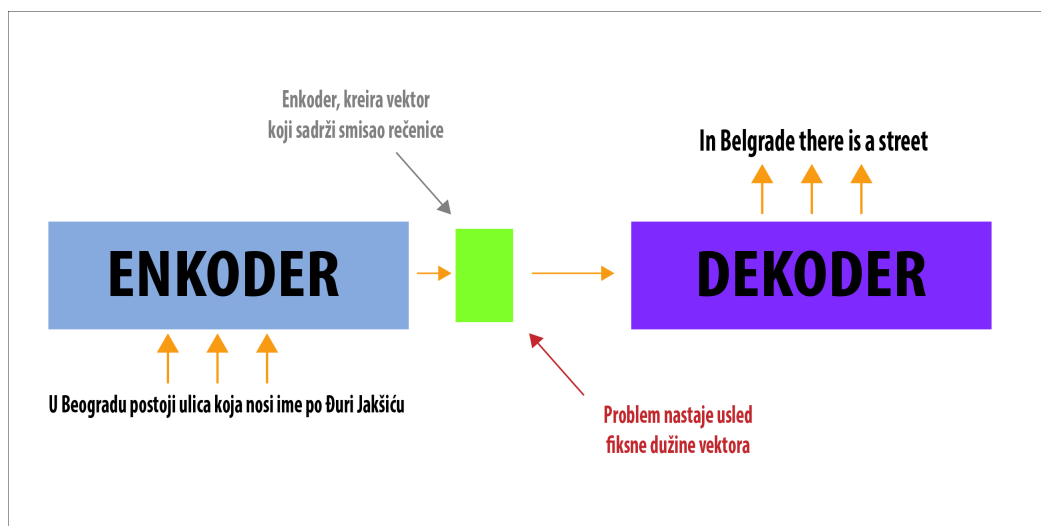


Slika 3.4: Primer strukture modela zasnovanog na sekvencama

Dekoder invertuje dejstvo enkodera pretvarajući skriveno stanje enkodera u izlazne sekvence. Po uzoru na enkoder, dekoder prolazi kroz skrivene slojeve neuronske mreže stvarajući pritom svoja skrivena stanja. Rezultat poslednjeg sloja je rezultujući vektor koji se transformiše u odgovarajuće reči na prirodnom jeziku.

Iako se često koriste rekurentne neuronske mreže, to ne mora da bude obavezna tehnika na kojoj su ovi digitalni sagovornici zasnovani. Model zasnovan na sekvencama se loše ponaša ukoliko su sekvence previše dugačke jer mogu da dovedu do nepredvidivih ponašanja pri generisanju skrivenih stanja. Na primer, pretpostavimo da se modelom zasnovanim na sekvencama implementira mašinski prevodilac

(program koji služi za prevođenje prirodnih jezika). Ukoliko se prosledi rečenica „Ja volim cveće” dekodler prevodioca će izgenerisati rečenicu „I love flowers”. Međutim, zbog fiksnog vektora skrivenih stanja koje prosleđuje enkoder, duže rečenice neće biti do kraja prevedene kao što može da se vidi na slici 3.5.



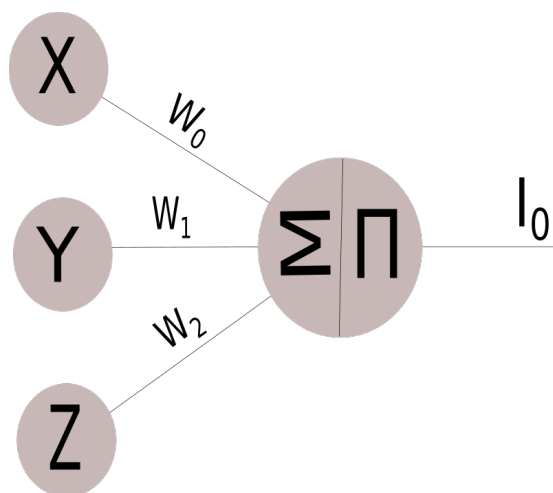
Slika 3.5: Mane modela zasnovanog na sekvencama

## Glava 4

# Neuronske mreže

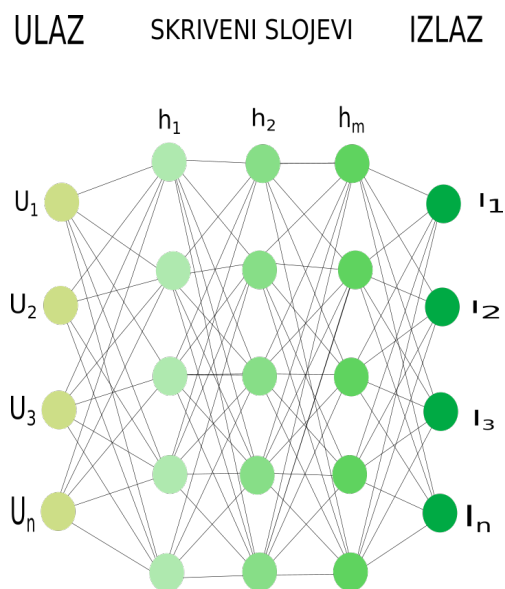
Pojava Interneta i kućnih računara krajem prošlog veka označila je početak napretka tehnologije i početak korišćenja velike količine podataka u korist zadovoljstva korisnika. Ta nova revolucija tehnologije je uticala na razvoj novih oblasti računarstva i na istraživanja u obradi velikih količina podataka. Kako bi se izvukle korisne informacije iz podataka koriste se metode mašinskog učenja, među kojima su i neuronske mreže. Osnovna ideja neuronske mreže je simulacija velike količine gusto napakovanih nervnih ćelija koje su međusobno povezane čime je omogućeno učenje pojmova, prepoznavanje objekata i donošenje odluka po uzoru na čoveka. Neke od primena uključuju: kategorizaciju teksta, prepoznavanje objekata na slici, autonomnu vožnju, digitalne sagovornice i druge. U opštem smislu, neuronske mreže predstavljaju matematičke modele čija struktura i funkcionalnost simulira strukturu i funkcionalnost ljudskog mozga. Simulacija ovako kompleksnog ljudskog organa dovodi do odličnih rezultata prilikom obučavanja mreža, što se vidi u njihovoj raznovrsnoj primeni u svakodnevnom životu. Međutim, neuronske mreže bi trebalo da se razmatraju samo ukoliko je na raspolaganju velika količina podataka jer male količine podataka često vode preprilagođavanju modela.

Sastavni deo neuronske mreže čine neuroni ili jedinice koje se sastoje od ulaza (koji je najčešće kombinacija vektora vrednosti i težinskog vektora), tela jedinice (u terminima veštačkog neurona, ulogu tela preuzima aktivaciona funkcija) i izlaza (slika 4.1). Za obradu velikih i kompleksnih podataka, jedna jedinica nije od velikog značaja. Radi postizanja boljeg efekta prilikom obrade podataka, koriste se udruženi neuroni, odnosno slojevi. Tipična neuronska mreža može imati i do milion neurona organizovanih u slojeve i broj slojeva ukazuje na složenost neuronskih mreža.



Slika 4.1: Model neuronske mreže sa jednim neuronom

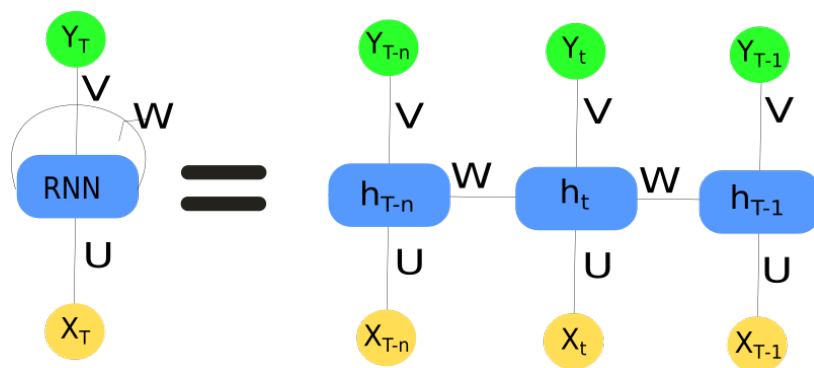
Svaka neuronska mreža sadrži početni sloj koji prihvata ulaze modela i proizvodi izlaz koji se prosleđuje drugom sloju koji može da bude izlazni ili skriveni sloj. Poslednji sloj u nizu proizvodi izlaz modela (slika 4.2). Način organizacije slojeva i njihovo povezivanje može da odredi vrstu neuronske mreže. Neke od njih uključuju mreže sa propagacijom unapred, potpuno povezane neuronske mreže, konvolutivne neuronske mreže i rekurentne neuronske mreže.



Slika 4.2: Ilustrovani prikaz neuronske mreže

## Rekurentne neuronske mreže

Sekvencijalni podaci kao što su, na primer, vremenske serije, ljudski govor, rečenice se obrađuju rekurentnim neuronskim mrežama [9]. Veza između ovih različitih vrsta podataka je činjenica da trenutno razmatrani podatak zavisi od svojih prethodnika. Na primer, glagol *eats* nema značenje ukoliko se posmatra samostalno u rečenici, međutim, možemo da pretpostavimo da njegovo značenje zavisi od neke reči koja je prethodno obrađena. Razmatranjem ovog problema postavljaju se i pitanja koliko reči prethodi trenutnoj i koje od njih su relevantne za obradu trenutne reči. Često se kaže da rekurentne mreže imaju memoriju zbog toga što slojevi dobijaju informacije iz prošlosti i prave vezu sa trenutnim informacijama. Na slici 4.3 se nalazi prikaz potpune rekurentne mreže (eng. fully recurrent neural networks). Sa leve strane se nalazi opšti prikaz rekurentne mreže dok se sa desne strane nalazi razmotana verzija rekurentne mreže.



Slika 4.3: Ilustrovana rekurentna neuronska mreža

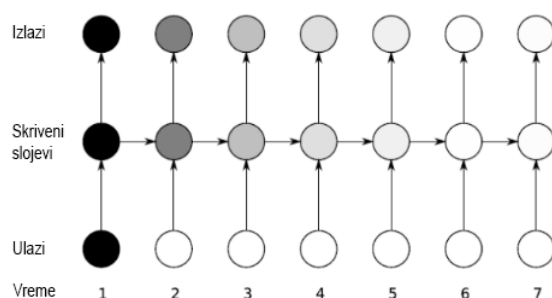
Sekvencione informacije se čuvaju u skrivenim stanjima mreža i pri svakom prolasku kroz slojeve (jedinice), informacije se iznova upotrebe čime se odaje utisak „čuvanja” informacija u dubljim slojevima mreže. Ovakvom tehnikom prosleđivanja podataka se traže korelacije podataka koje su „vremenski” udaljene. Osnovna karakteristika RNN strukture jeste predstavljanje istorije koja obuhvata teorijski sve prethodne reči i do te strukture se stiže obučavanjem. U skladu sa tim, modeli jezika bazirani na rekurentnim neuronskim mrežama mogu, bar u teoriji, uočiti i pravilnosti koje se manifestuju na veoma dugim sekvencama reči, čak i van granica pojedinih rečenica. Eksperimenti su potvrdili da RNN modeli jezika po uspešnosti i modelovanja jasno prevazilaze mogućnosti ostalih modela neuronskih mreža [10]. Velika prednost RNN-a u odnosu na ostale vrste neuronskih mreža je ta što dozvoljava

da ulazni podaci, odnosno sekvence, budu proizvoljne dužine. Ulazi mreže mogu da predstavljaju odgovarajuće kodiranje jedne reči, dok skrivena stanja održavaju smisao rečenice koja se menja prilikom obrađivanja novih reči. Često se u slučaju kategoričkih podataka koristi binarno kodiranje koje bi uvelo dodatnu kompleksnost kod drugih mreža ali RNN sa „lakoćom” obrađuje i ovaj slučaj. Međutim, iako se veruje da RNN „sadrži” memoriju ona je kratkotrajna i nemoguće je izvući korisne informacije iz dugačkih rečenica.

Neke od mana RNN-a su [9]:

Eksplozivajući gradijenti i nestajući gradijenti: usled velikog broja operacija prilikom računanja gradijenata, može se desiti prekoračenje ili potkoračenje. Ekstremne vrednosti gradijenta (0 ili  $\infty$ ) loše utiču na vrednosti težina prilikom učenja.

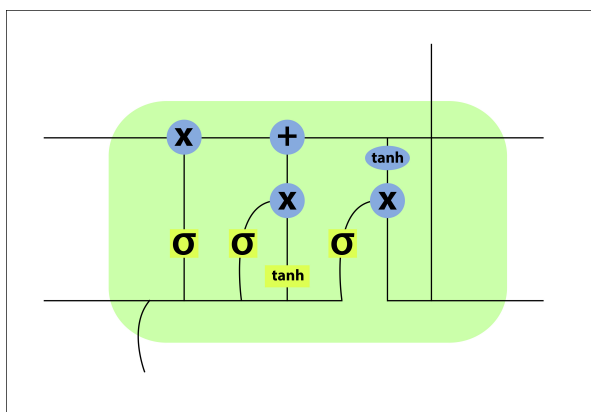
Gubitak bitnih informacija iz prošlosti: jedna od vodećih karakteristika RNN-a je njihova sposobnost povezivanja trenutnih i prethodnih informacija, međutim, u mnogim slučajevima njihovu sposobnost nije moguće iskoristiti. Na primer, pretpostavimo da je zadatak modela da predvidi sledeću reč na osnovu njenih prethodnika. Ukoliko je potrebno da se predvidi poslednja reč u rečenici „*Mrizantema je cvet*” nisu potrebne dodatne informacije osim reči *Mrizantema*. Sa druge strane postoje slučajevi koji zahtevaju dublji kontekst. Na primer, „On je detinjstvo proveo u Rusiji... i tečno govori ruski.”, reč *ruski* očigledno zavisi od reči *Rusija* ali udaljenost ove bitne informacije i trenutne reči je previše velika. Na slici 4.4 se može videti ilustrovani primer gubitka informacija.



Slika 4.4: Problem gubljenja informacija iz prošlosti

## Duga kratkoročna memorija

Glavni problem rekurentnih neuronskih mreža predstavlja njihova zavisnost od dužine ulazne sekvence koja ima za posledicu nemogućnost dugoročnog čuvanja podataka kod velikih sekvenci. Mane koje imaju obične rekurentne mreže mogu da se prevaziđu uz pomoć posebne vrste rekurentne mreže, a to je duga kratkoročna memorija ili skraćeno LSTM (eng. Long Short-Term Memory) [7].



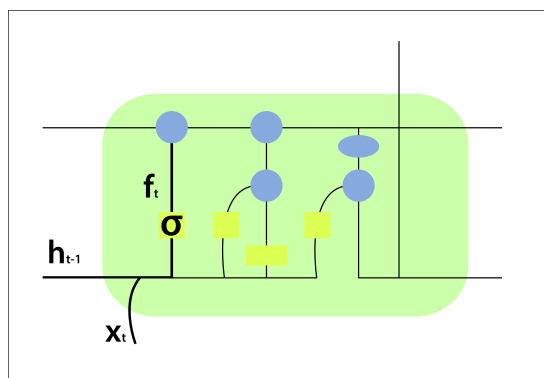
Slika 4.5: LSTM ćelija

Moć LSTM mreže leži u unutrašnjoj strukturi koja se sastoji od kapija (eng. gate) čija je uloga da odlučuju koje sekvence su vredne pamćenja čime se ostvaruje čuvanje podataka u dubljim slojevima mreže (slika 4.5). Za razliku od običnih RNN-mreža koje imaju samo jednu aktivacionu funkciju u jednom neuronu, LSTM ima nekoliko aktivacionih funkcija i svaka zasebno formira jednu kapiju. Ovakvim ponašanjem mreža može da uči duži niz epoha i time se stvara veza između prošlih događaja sa njihovim posledicama u sadašnjosti. Kapije se aktiviraju prilikom prihvatanja ulaznih podataka i propuštaju (odnosno blokiraju) podatke na osnovu svojih skupova težina koje se računaju prilikom obučavanja mreže. Tok kretanja podataka kroz mrežu je sledeći.

### Kapija zaboravljanja

Zadatak ove kapije je odabir informacija koje će se proslediti, a koje neće (slika 4.6). Informacija iz prethodnog skrivenog stanja jedinice i trenutnog ulaza jedinice se prosleđuju aktivacionoj funkciji, najčešće sigmoidnoj funkciji koja prosleđene vrednosti preslikava u interval  $(0, 1)$ . Vrednost dobijena aktivacijom funkcije određuje koliki će biti uticaj prethodnog stanja ćelije na trenutno stanje, odnosno kvantifikuje se verovatnoća čuvanja informacija u memoriji.

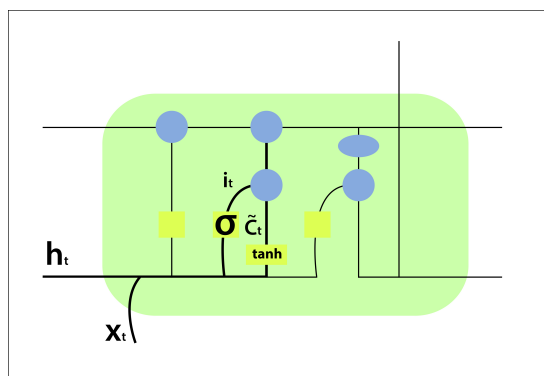




Slika 4.6: Kapija zaboravljanja

### Ulazna kapija

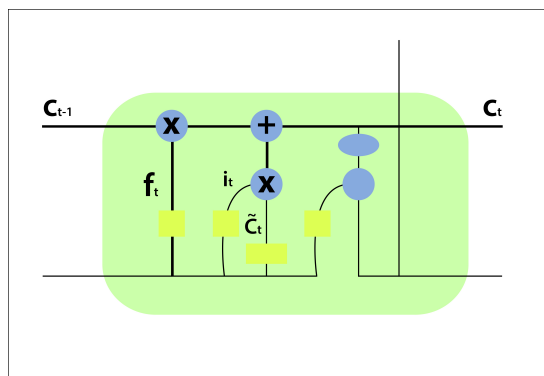
Zadatak ove kapije je ažuriranje stanja jedinice. Trenutni ulaz jedinice i prethodno skriveno stanje jedinice se prosleđuju zajedno ovoj kapiji koja sadrži dve aktivacione funkcije, tangens hiperbolički i sigmodnu funkciju. Jedna funkcija preslikava vektore u brojeve iz intervala  $(0, 1)$ , dok druga u brojeve iz intervala  $(-1, 1)$ . Aktivacijom sigmoidne funkcije, kao i u slučaju kapije zaboravljanja, dobija se uticaj prethodnog skrivenog stanja, dok se aktivacijom funkcije tangens hiperbolički dobija vektor potencijalno novih kandidata (Slika 4.7). Rezultati ovih funkcija, kao i vrednosti kapije zaboravljanja se dalje kombinuju kako bi se dobilo novo stanje ćelije.



Slika 4.7: Ulazna kapija

## Ćelija

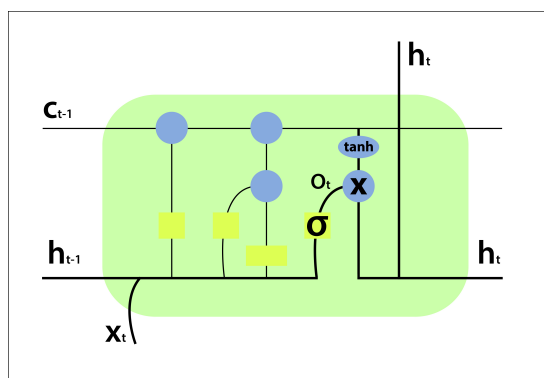
Stanje ćelije<sup>1</sup> prihvata kao ulaz prethodno stanje ćelije koje se vektorski množi sa rezultujućim vektorom kapije zaboravljanja. Na rezultat se dodaje rezultujući vektor ulazne kapije čime se dobija novo stanje ćelije (Slika 4.8). Srž LSTM neuronske mreže se nalazi u znaku + koji rešava problem računanja gradijenata.



Slika 4.8: Stanje ćelije

## Izlazna kapija

Zadatak ove kapije je izračunavanje skrivenog stanja ćelije. Kao i kod ulazne kapije, koriste se vrednosti dobijene aktivacijom funkcije tangens hiperboličkog trenutnim stanjem ćelije i vrednosti dobijene aktivacijom sigmoidne funkcije trenutnim ulaznim podacima. Potom se rezultati funkcija množe čime se dobija skriveno stanje trenutne ćelije (Slika 4.9).



Slika 4.9: Izlazna kapija

<sup>1</sup>Radi lakšeg shvatanja LSTM, govori se samo o jednoj LSTM ćeliji.

Iako prevazilazi mane koje rekurentne mreže imaju LSTM čuva informacije iz prošlosti jer se podaci čitaju jedan po jedan, tj. postoji poredak prošlog i trenutnog podatka. Ponekad je, radi preciznijeg predviđanja, potrebno znati i buduće podatke. Modifikacija se vrši dodavanjem još jednog sloja koji predstavlja inverz rečenice prosleđene mreži. Takav LSTM se naziva bidirekcion (eng. *bidirectional*) ili dvosmerni LSTM. Kombinovanjem ovih slojeva mreža može preciznije da procesira rečenicu i relaciju između reči.

### Model zasnovan na sekvencama

Model zasnovan na sekvencama je tehnika mašinskog učenja koja predstavlja transformaciju jedne sekvence u drugu. Sekvenca predstavlja niz srodnih podataka kao što su reči ili brojevi koji predstavljaju vremenske serije. Modeli mogu da se kategorizuju prema dužini ulazne i izlazne sekvence (dužina je ili fiksna ili promenljiva) ili tipu podatka ulaznih i izlaznih sekvenci (istih ili različitih tipova). Često se koristi LSTM mreža ili GRU (eng. *Gated Recurrent Unit*) kako bi se izbegle mane RNN-a. Arhitektura koja se koristi za ovakvu tehniku je enkoder – dekode arhitektura. Često se poistovećuju sa autoenkoderima koji mogu da se podele na enkoder i dekode ali zapravo predstavlja jednu mrežu, dok model zasnovan na sekvencama sadrži dve nezavisne povezane mreže (enkoder i dekode). Obe mreže su rekurentne mreže, ali zadaci koje one obavljaju su veoma različiti. Enkoder prima sekvencu iz ulazne rečenice i procesira jednu reč u svakom trenutku vremena. Slojevi enkodera prave vektor konteksta ili „misao rečenice” koji se prosleđuje dekoederu. Ovaj vektor se naziva i misaoni vektor jer prenosi kontekst (misao) prosleđene rečenice i fiksne je dužine. Slojevi dekoedera tumače vektor konteksta i nakon prolaska kroz mrežu generišu binarnu matricu, na osnovu koje se formiraju vektori čije se vrednosti nalaze u rečniku modela. Dužine tih sekvenci koje se primaju i dobijaju ne moraju nužno da budu iste, ali što su rečenice na ulazu duže teže je napraviti kompaktniji vektor konteksta koji utiče na izlaz.

Postoje dva glavna problema u vezi sa celokupnom enkoder–dekode arhitekturom:

- Prvi problem predstavlja odnos dimenzije mreže i dužine ulaznih rečenica. Na primer, ukoliko su dimenzije mreže male u odnosu na dužinu rečenica onda se neophodne informacije iz rečenica ne čuvaju u misaonom vektoru. Sa druge

strane ukoliko su dimenzije mreže prevelike u odnosu na dužinu rečenice onda se nepotrebno troši velika količina resursa za čuvanje malog broja informacija.

- Drugi problem je nemogućnost čuvanja velike količine informacija u misaonom vektoru koji je fiksne dužine.

## Mehanizam pažnje

Mehanizam pažnje (eng. attention mechanism) je komponenta mreže koja teži da kvantifikuje međuzavisnost između ulaznih vektora. Ideja je da se bitne reči označe kao važne tako što se tim rečima dodeli veća težina. Posmatrajući model zasnovan na sekvencama, enkoder prolazi kroz ulaznu sekvencu i transformiše je u misaoni vektor fiksne dužine, odbacujući pritom sva među stanja enkodera. Vektor se kasnije prosleđuje ulazu dekodera kao početak generisanja izlaza.



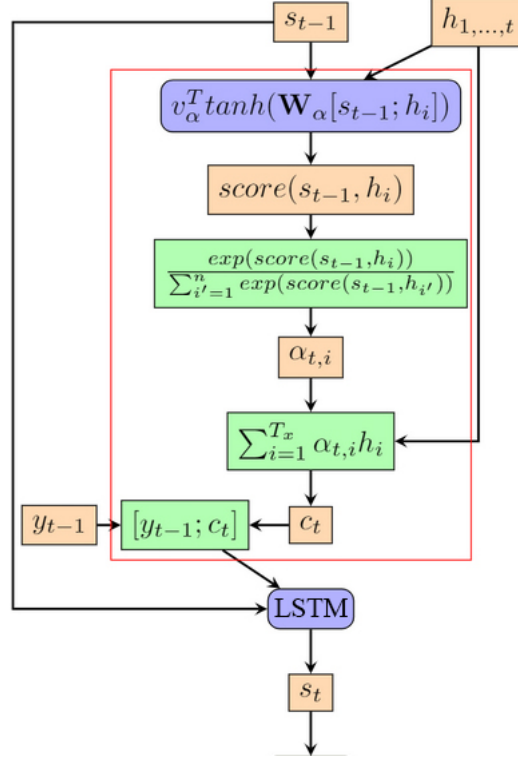
Slika 4.10: Ideja primene mehanizma pažnje

Za kraće sekvence je ovaj način transformisanja i generisanja sekvenci veoma dobar ali sa povećanjem dužine sekvenci (rečenica) implementacija misaonog vektora kao vektora fiksne dužine počinje da pravi usko grlo (eng. bottleneck) i nemoguće je izvući „misa“ u jedan vektor. Ideja je da se uspostavi jedinstvena veza između svakog izlaza iz dekodera ka svim skrivenim stanjima enkodera. Tako izlaz iz dekodera može da pristupa celokupnom skupu ulaznih sekvenci i može da „probere“ podatke iz neke konkretne sekvence i prosledi ih izlazu. Opšta ideja mehanizma pažnje se može videti na slici 4.10.

Postoji nekoliko načina implementacije mehanizma pažnje, među kojima su najpoznatiji Luongov (eng. Luong attention mechanism) i Bahdanaov (eng. Bahdanau attention mechanism) mehanizam pažnje [3].

U modelima zasnovani na sekvencama, bez mehanizma pažnje, dekoderi se obučavaju da predvide izlaz  $y_t$  koristeći misaoni vektor  $c$  i reči koje su prethodno predviđene. Važni parametri koji se koriste prilikom računanja su:  $x$  koji predstavlja

ulazni vektor enkodera,  $y$  koji predstavlja izlazni vektor dekodera,  $s$  koji predstavlja vektor skrivenih stanja dekodera,  $h$  koji predstavlja vektor skrivenih stanja enkodera. Ilustrativan primer rada mehanizma pažnje može se videti na slici 4.11.



Slika 4.11: Ilustrovani primer koraka primene mehanizma pažnje

Mehanizmom pažnje misaoni vektor  $c$ , koji dekodер koristi za predviđanje (odnosno generisanje izlazne reči), transformiše u vektor (sekvencu) misaonih vektora  $c_i$ . Novouvedene promenljive se koriste kako bi se izračunalo skriveno stanje dekodera  $s_t$ . Kako bi se izračunala vrednost jednog misaonog vektora  $c_t$  koristi se vrednost poravnanja (eng. alignment score) koja predstavlja kvantifikovanu vezu između ulaza  $x_i$  i izlaza  $y_t$ , odnosno koliko vrednost  $x_i$  odgovara vrednosti  $y_t$  (izraz 4.3).

$$score(s_t, h_i) = v_{\alpha}^T \tanh(\mathbf{W}_{\alpha}[s_t; h_i]) \quad (4.1)$$

$$\alpha_{t,i} = \frac{\exp(score(s_{t-1}, h_i))}{\sum_{k=1}^n \exp(score(s_{t-1}, h_k))} \quad (4.2)$$

$$c_t = \sum_{i=1}^{T_x} \alpha_{t,i} h_i \quad (4.3)$$

Nakon ovih transformacija, skriveno stanje dekodera  $s_t$ , misaoni vektor  $c_{t+1}$  i predviđeni izlaz dekodera  $y_{t+1}$  se prosleđuju LSTM mreži dekodera i vrše se aktivacije opisane u delu rada koji opisuje LSTM mrežu.

# Glava 5

## Implementacija

U ovoj glavi je prikazan model neuronske mreže koja služi kao osnova jednog inteligentnog digitalnog sagovornika otvorenog domena. Baza podataka koja će biti korišćena za obučavanje mreže je skup dijaloga sa Univerziteta Kornel u Njujorku koji sadrži kolekciju izmišljenih razgovora iz sirovih filmskih scenarija [5]. Podaci koji se koriste za obučavanje modela su tekstualnog tipa i njihovo preprocesiranje uključuje transformaciju reči u vektore brojeva i obradu nepostojećih vrednosti. Nakon obrade podaci se prosleđuju neuronskoj mreži radi njenog obučavanja. Za potrebe implementacije neuronske mreže korišćena je platforma *Google Cloud* gde je dostupno 64 GB RAM memorije i virtuelni CPU *Intel Xeon Scalable Processor (Cascade Lake) 2nd Generation*. Model predstavlja neuronsku mrežu sastavljenu od dve rekurentne neuronske mreže, enkodera i dekodera. Jezik koji je korišćen za implementaciju je Python i implementacija modela je javno dostupna na Github platformi [8].

### 5.1 Preprocesiranje podataka

Često se pri izradi digitalnih sagovornika bira tema koja je vezana za neku određenu delatnost, kao što su kupoprodaja ili informativni digitalni sagovornici. Rečnik digitalnih sagovornika ovakvog specifičnog domena nije veliki, ali sagovornici koji se prave za opštu upotrebu moraju da imaju velike rečnike koje će pokrivati većinu reči koje se pojavljuju u dijalogima. Model koji se u ovom radu koristi se obučava nad skupom koji se sastoji od 221616 podataka (broj označava broj parova u relaciji pitanje–odgovor). Podaci koji se dobijaju na početku su puni nedostataka koji umanjuju kvalitet modela, ali i kvalitet učenja. Značajan proces metoda mašinskog

učenja je proces preprocesiranja podataka jer kvalitet izlaznih rezultata (i tok učenja) umnogome zavisi od toga kako su podaci pripremljeni. Svrha preprocesiranja je da podaci, u ovom slučaju tekst, budu u formi koja je pogodnija za analizu i obradu. Prilikom preprocesiranja podataka za obučavanje neuronske mreže, koja će biti osnova za dizajniranje digitalnog sagovornika, konstruiše se rečnik samog sagovornika. Uloga rečnika se najviše ističe prilikom konstruisanja programa za mašinsko prevođenje jezika (eng. machine translation). Na primer, kako bi se model istrenirao da prevodi sa srpskog na engleski jezik potreban je rečnik koji će predstavljati vezu reči na srpskom i odgovarajuće reči na engleskom jeziku (u praksi se rečnici konstruišu drugačije ali ideja ostaje ista). Za preprocesiranje teksta koriste se metode obrade prirodnih jezika kao što su filtriranje sadržaja rečenica (izbacivanje nekih reči koje su rezervisane za format čuvanja teksta kao što su HTML delovi stranice), problem interpunkcije (u ovaj problem spadaju i redni brojevi, decimalni brojevi, skraćenice, inicijali, emotikoni i drugo), tokenizacija (podela reči na smislene reči). Prilikom konstrukcije modela ovog rada korišćeni su sledeći koraci.

### **Transformacija velikih u mala slova**

U mnogim konverzijama često nastanu greške prilikom kucanja koje uključuju mala i velika slova, tako *Australija* može da se napiše kao *australija* ili *AuStRaLiJa*. Pošto model ne razume da je to sve ista reč, svakoj reči biće dodaljen drugačiji broj i biće tretiran drugačije. Kako bi se izbegla zabuna oko ovakvih situacija pribegava se transformaciji velikih slova u mala. Iako je pogodno za pretragu ponekad je potrebno zadržavanje velikih slova kao što su vlastite imenice. Ipak, ova vrsta transformacije ima više pozitivnih strana nego negativnih.

### **Tokenizacija**

Predstavlja proces izdvajanja reči u rečenici ili korpusu. Na primer, tokeni rečenice „ovo je primer tokenizacije” je vektor reči [„ovo”, „je”, „primer”, „tokenizacije”]. Međutim, ako se doda razmak pre i posle znakova interpunkcije, to će učiniti znakove interpunkcije posebnim celinama koje treba izdvojiti i nemaju neko specijalno značenje (na primer, izdvajanjem tokena iz rečenice „Zdravo , kako si ?” dobija se vektor [„Zdravo”, „,”, „kako”, „si”, „?”]). U specijalne slučajeve spadaju i skraćenice koje često uključuju tačke i ustaljene reči koje uključuju apostrofe.

### **Izbacivanje znakova interpunkcije**

Znakovi interpunkcije nemaju značenje kao tokeni jer je njihova uloga u rečenici da razdvoje bitne delove rečenice i često se izbacuju. Međutim, u znakove interpunkcije spadaju i navodnici koji razdvajaju ključne reči od ostatka rečenice ili su i



sami deo zapisa reči kao što su *isn't* ili *'70*. Na primer, u rečenicama na engleskom jeziku reči *is not* predstavljaju jedinstvenu celinu koja se često piše u naučnim dokumentima ali u svakodnevnom pisanju (govoru) se upotrebaljava i manje formalan oblik *isn't*. Za mašinu, ili konkretno digitalnog sagovornika, ova dva oblika nemaju isto značenje i iz tog razloga je poželjno da se koristi jedinstveni oblik i zamene se sva pojavljivanja reči *isn't* u *is not*. Zamene ovakve vrste se najčešće obavljaju regularnim izrazima koji su pogodni za traženje i menjanje onih delova rečenice koji zadovoljavaju uslov filtriranja.

Primenom prethodno pomenutih transformacija dobijaju se sledeći rezultati:

```
[
  "Then_that's_all_you_had_to_say.",
  "I_don't_want_to_know_how_to_say_that_though.",
  "Right._See?_You're_ready_for_the_quiz."
]
[
  ['then', 'that', 'is', 'all', 'you', 'had', 'to', 'say'],
  ['i', 'do', 'not', 'want', 'to', 'know', 'how', 'to', 'say', 'that', ' ',
   'though'],
  ['right', 'see', 'you', 'are', 'ready', 'for', 'the', 'quiz']
]
```

Neuronske mreže obično ne podržavaju podatke koji nisu numeričkog tipa i iz tog razloga se konstruiše rečnik koji predstavlja preslikavanje reči u indeks, odnosno broj. Postoji dosta pristupa konstruisanju rečnika, ali izbor pristupa neće uticati na ponašanje modela prilikom obučavanja jer je glavni cilj svakog pristupa da se svaka reč preslika u jedinstveni broj. Najjednostavniji pristup konstrukciji rečnika je inkrementalni pristup koji podrazumeva postojanje vektora reči (odnosno tokena koji su izdvojeni iz rečenica) i svakoj reči se dodeli broj koji je za jedan veći od broja svog prethodnika, čime je ostvaren uslov jedinstvenosti.

```
'i' : 25
'thought' : 26
'would' : 27
'start' : 28
'with' : 29
'pronunciation' : 30
'if' : 31
'that' : 32
'is' : 33
'okay' : 34
```

Podaci koji se dobijaju su vektori brojeva čija je dužina jednaka dužini odgovarajuće rečenice, a elementi predstavljaju indeks odgovarajuće reči koja se pojavljuje u rečenici. Dodatno se u rečnik ovog skupa pridružuju četiri podrazumevana tokena kojima se dodeljuje specijalno značenje i rezervisani indeksi (najčešće su to 0, 1, 2 i 3). To su *PAD*, *BOS*, *EOS* i *UNK* koji redom označavaju punjenje (eng. Padding), početak rečenice (eng. Beginning of sentence), kraj rečenice (eng. End of sentence) i nepoznato (eng. Unknown).

```
'PAD' : 0
'BOS' : 1
'EOS' : 2
'UNK' : 3
```

Uloga ovih tokena je: označavanje početka i kraja sekvence (*BOS* i *EOS*), označavanje nepoznatih reči (*UNK*) i dopunjavanje manjih sekvenci do predefinisane dužine (*PAD*). Određivanjem dužine rečenice, odnosno vektora indeksa, zadaje ograničenje rečenicama koje će biti prosleđene modelu. Moguće je zadati donje i gornje ograničenje rečenicama i one rečenice čije su dužine manje od gornjeg ograničenja se dopunjuju specijalnim tokenom *PAD*. Rečenice koje imaju ulogu odgovora sagovornika, predstavljaju posebnu vrstu vektora jer se dopunjuju tokenima *BOS* i *EOS* čime se naznačava početak i kraj rečenice. U sledećem primeru može se videti kako se zadata rečenica (koja ima 7 reči) transformiše u vektor dužine 13 sa tokenima *BOS*, *EOS* i *PAD*.

```
"Then_that's_all_you_had_to_say."
['then', 'that', 'is', 'all', 'you', 'had', 'to', 'say']
['BOS', 'then', 'that', 'is', 'all', 'you', 'had', 'to', 'say', 'EOS']
[1, 421, 241, 23, 90, 12, 901, 25, 1231, 2]
[1, 421, 241, 23, 90, 12, 901, 25, 1231, 2, 0, 0, 0]
```

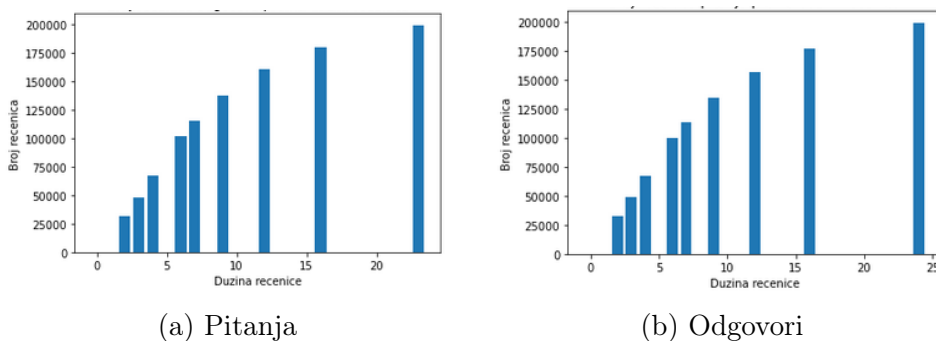
U zavisnosti od dužina rečenica koje će se koristiti za obučavanje nastaju rečnici različitih veličina i sadržaja. Smanjenjem veličine rečnika nastaju veliki propusti prilikom pravljenja vektora koji se prosleđuju modelu, tj. nastaje veliki broj nepoznatih reči. Broj nepoznatih reči je u korelaciji sa veličinom rečnika koji se izgeneriše, što je rečnik veći broj nepoznatih reči je manji i u idealnoj situaciji sve reči su poznate. Zbog ograničenosti memorije veličina rečnika mora da se smanji i jedna mogućnost je da se nasumično izabere veličina rečnika ne obazirući se na reči. Ovaj postupak može da dovede do razređenosti podataka koji će se proslediti modelu. Suprotan pristup podrazumeva čuvanje onih reči koje su učestale (neke reči se pojavljuju hiljadu puta dok se neke pojavljuju samo jednom, uvođenjem praga

izbacuju se one reči koje ne zadovoljavaju zadati prag pojavljivanja), čime se dobija veza veličine rečnika i zadatog praga najmanjeg pojavljivanja reči (eng. threshold). Na slici 5.1, može se videti veza nepoznatih reči i praga najmanjeg pojavljivanja reči. Sa porastom praga broj nepoznatih reči postaje veći, a veličina rečnika se smanjuje.



Slika 5.1: Odnos broja nepoznatih reči i praga

Na veličinu rečnika utiče i dužina rečenica koja se koristi, gde dužina predstavlja broj reči u rečenici (slika 5.2). Najduža rečenica među dijalozima ima 563 reči, što memorijski može da bude veoma zahtevno i pošto se rečenice dopunjuju tokenima *PAD* (koji se najčešće preslikavaju u 0 zbog pogodnih funkcija koje mogu da se iskoriste prilikom implementacije) dobijaju se razređene matrice podataka. Problem odabira dužine rečenice koje će se koristiti zavisi od svrhe modela i količine dostupne memorije.



Slika 5.2: Broj rečenica prema maksimalnoj dužini rečenica

Još jedna metoda koja se koristi prilikom obrade podataka je binarno kodiranje, odnosno predstavljanje kategoričkih podataka (nenumeričkih podataka) preko kombinacija binarnih vektora. Mnoge metode mašinskog učenja ne funkcionišu najbolje

sa nenumeričkim podacima kao što su, na primer, slova, reči, datumi<sup>1</sup>. Razlog je matematička priroda algoritama mašinskog učenja i ova osobina predstavlja ograničenje algoritama mašinskog učenja. Rečenice koje se transformišu u vektore indeksa sadrže brojeve koji mogu biti veliki ili mali. Međutim, vrednosti tih brojeva ne smeju da utiču na obučavanje modela (na primer, broj 1000 je veći od 23, ali ne znači da je reč sa indeksom 1000 bitnija od reči sa indeksom 23). Predstavljanje tih brojeva putem binarnih vektora omogućuje ravnopravnost podataka prilikom obučavanja. Problem koji se javlja kod ove tehnike je velika dimenzionalnost dobijene matrice, što dovodi do upotrebe dodatne memorije i dodatnog vremena obučavanja modela. Zbog ograničene memorije koju platforma pruža korišćen je manji broj rečenica veće dužine kako bi se prikazao rad mehanizma pažnje i relativno mali rečnik kako bi se smanjila dimenzionalnost binarne matrice.

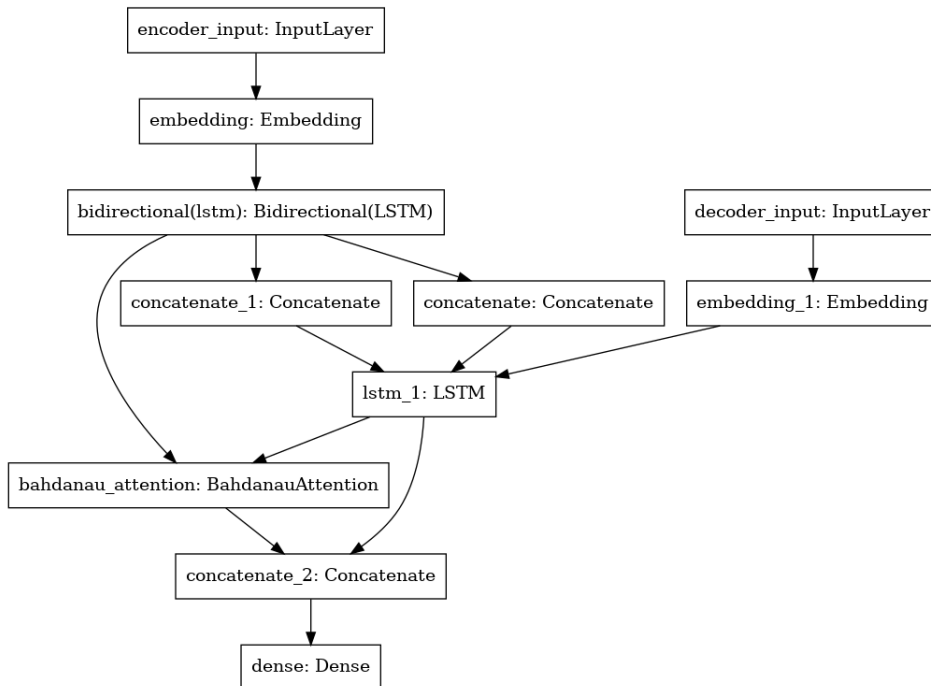
## 5.2 Podešavanje parametara neuronske mreže

Model koji se koristi u ovom radu je nastao kao kombinacija dva različita modela koje imaju ulogu enkodera i dekodera i koji u svojoj osnovi predstavljaju rekurentne neuronske mreže. Za konstrukciju neuronskih mreža korišćene su pogodnosti Keras i Tensorflow biblioteka. Enkoder mreža se sastoji od ulaznog sloja (eng. input layer), sloja ugrađivanja (eng. embedding layer) i dvosmernog LSTM sloja. Dekoder se sastoji iz ulaznog sloja, sloja ugrađivanja i običnog LSTM sloja. Izlaz i skrivena stanja dvosmernog LSTM-a se prosleđuju običnom LSTM sloju dekodera koji ujedno predstavlja spoj enkodera i dekodera. Svaki LSTM sloj sadrži parametar odsecanja (eng. dropout) koji predstavlja metaparametar regularizacije i teži da umanjí mogućnost pojave preprilagođavanja modela. Parametar odsecanja određuje verovatnoću odbacivanja neurona prilikom ažuriranja matrica težina. Krajnji rezultat njihovih aktivacija se prosleđuje sloju koji ima ulogu mehanizma pažnje. Mehanizam pažnje u svojoj strukturi predstavlja jednu rekurentnu mrežu čije se težine ažuriraju putem formula koje su obrađene u glavi 4. Težine rekurentne mreže mehanizma pažnje se prilikom obučavanja inicijalizuju nasumično odabranim brojevima (odnosno matricama). Rezultat koji se dobija primenom mehanizma pažnje se prosleđuje potpuno povezanom sloju (eng. dense) koji računa izlaz koristeći funkciju *softmax*. Arhitektura mreže može se prikazati u obliku dijagrama kao na slici 5.3 i

---

<sup>1</sup>Neke metode mašinskog učenja, kao što su drvo odlučivanja, mogu da funkcionišu i sa kategoričkim podacima

u obliku tabele 6.1.



Slika 5.3: Ilustrovana struktura modela generisana uz pomoć jezika Python <sup>2</sup>

Dodavanjem slojeva modelu dimenzija mreže se povećava što uzrokuje potrebu za dužim vremenom i većim prostorom za obučavanje mreže. Zbog tehničkih ograničenja nemoguće je model sa velikim brojem parametara obučiti sa dugačkim rečenicama i iz tog razloga se parametrima koji se odnose na bazu podataka digitalnog sagovornika posvećuje najveća pažnja. Parametri koji se razmatraju prilikom obučavanja modela i koji se tiču baze podataka su:

**dužine rečenica pitanja i odgovora:** odabrana dužina za rečenice koje odgovaraju pitanjima je 20 reči u rečenici, dok za razliku od pitanja odabrana dužina za rečenice koje predstavljaju odgovore je 18. Razlog različite dužine rečenica je dodavanje tokena *BOS* i *EOS* na početak i kraj odgovora čime se ostvaruje maksimum od 20 reči.

**broj instanci parova rečenica:** ukoliko je odabrana dužina rečenice 20 dobija se 160060 parova pitanja i odgovora ali nije moguće sve parove iskoristiti

---

<sup>2</sup>Struktura modela je prikazana u obliku grafa povezanosti koji je generisan funkcijom `plot_model` koji se nalazi u modulu `keras.utils`.

Tabela 5.1: Prikaz slojeva mreže

Sloj (tip)	Ulaz je dobijen iz izlaza sloja:
encoder_input (ulazni sloj)	
embedding (sloj ugrađivanja)	encoder_input
decoder_input (ulazni sloj)	
bidirectional (dvosmerni sloj)	embedding
embedding_1 (sloj ugrađivanja)	decoder_input
concatenate (konkatenacija skrivenih stanja dvosmerne mreže)	bidirectional[0][1] bidirectional[0][3]
concatenate_1 (konkatenacija skrivenih stanja dvosmerne mreže)	bidirectional[0][2] bidirectional[0][4]
lstm_1 (LSTM)	embedding_1 concatenate concatenate_1
bahdanau_attention (sloj pažnje)	bidirectional lstm_1
concatenate_2	lstm_1 bahdanau_attention
dense (potpuno povezan sloj)	concatenate_2

za obučavanje zbog ograničene memorije koju platforma pruža. Zato se koristi 50000 parova pitanja i odgovora za obučavanje modela.

**optimalan prag najmanjeg pojavljivanja reči u skupu:** reči koje se koriste za konstrukciju rečnika se biraju ukoliko zadovoljavaju dati prag pojavljivanja. Za konstrukciju ovog rečnika koristi birale su se one reči koje se pojavljuju najmanje 12 puta. Time se postižu i neželjene posledice nedostajućih vrednosti ali se dobija na memoriji koja je potrebna za obučavanje modela.

**veličina rečnika koji se koristi:** iako nije savršena, veličina rečnika je 7183 koji odgovara broju poznatih reči u rečniku.

Skup podataka je podeljen na tri dela, skup za obučavanje, skup za validaciju i skup za testiranje (razmera ovih skupova je 60:20:20). Za obučavanje je korišćeno 200 epoha sa 1024 paketa (eng. batches). Skupovi se određuju nasumično radi dobijanja raznovrsnijih podataka za učenje i validaciju. Model koji se dobija obučavanjem se kasnije testira nad skupom za testiranje kako bi se dobila ocena tačnosti predviđanja

modela. Skupovi za obučavanje i validaciju se prosleđuju modelu prilikom obučavanja. Prilikom obučavanja i testiranja modela, potrebno je pratiti performanse modela koje se ogledaju u merama evaluacije i funkciji greške.

### 5.3 Obučavanje i evaluacija modela

Keras biblioteka sadrži veliki broj metoda uz pomoć kojih je moguće obučiti model i razne metrike evaluacija koje se koriste za proveru preciznosti modela, među kojima su matrica konfuzije, f1-mera, srednjekvadratna greška. Često se za potrebe modele digitalnih sagovornika koriste drugačije mere evaluacije, među kojima se koriste mera tačnosti (eng. accuracy) predviđanja sekvenci i BLEU mera (eng. BLEU score). Tačnost predstavlja odnos broja tačno predviđenih sekvenci i ukupnog broja predviđenih sekvenci. Mera tačnosti se primenjuje nad skupovima za obučavanje, validaciju i testiranje. Međutim, tačnost nije najbolja mera evaluacije jer se generisane reči mogu razlikovati od pravih odgovora ali mogu predstavljati validan odgovor. Na primer, ukoliko se u bazi nalazi pitanje „Da li si lepo proveo dan?” i odgovarajući odgovor „Jesam, bio sam u parku” i ukoliko digitalni sagovornik izgeneriše odgovor „Jesam, a ti” ne može se reći da je izgenerisana rečenica pogrešna ali u odnosu na odgovor iz baze nije tačna. Stoga se koriste drugačije mere evaluacije među kojima je i BLEU mera, koja se često koristi za mašinsko prevodenje i digitalne sagovornike i ideja mere je da se prihvataju što približnije rečenice. BLEU mera je broj iz intervala (0, 1) koji se zasniva na kvantifikovanju razlike predviđene rečenice i ciljane rečenice. Poželjno je postići meru između (0.6 i 0.7) jer se u tom intervalu može kvantifikovati razgovor dva čoveka. Ukoliko se dobije mera koja je približno 1, model je prilagođen skupu za obučavanje.

Među bitnim odlikama modela je i funkcija gubitka (eng. loss function). Funkcijom gubitka se računa razlika predviđene vrednosti algoritma i stvarne vrednosti. Za implementaciju modela korišćena je funkcija kategoričke unakrsne entropije (eng. Categorical CrossEntropy) koja se koristi za kvantifikovanje razlike između dve funkcije raspodela verovatnoća (za proizvoljan skup događaja) i podrazumeva da su ciljane vrednosti binarno zapisane. Tačnost koja odgovara funkciji greške koja se koristi pri izradi modela je tačnost unakrsne kategoričke entropije koja računa koliko puta je tačno predviđena sekvenca rezultata u obliku binarnih matrica.

Prilikom obučavanja modela korišćen je ADAM optimizator (eng. adaptive moment estimation) koji predstavlja tehniku ažuriranja težina mreže [14]. Zasniva se

na upotrebi ocena prvog i drugog momenta gradijenta, koji predstavljaju brzinu i ubrzanje kretanja gradijenta kroz prostor, koji menjaju tok gradijentnog spusta zavisi u zavisnosti od regiona u kojem se funkcija nalazi. Bitan parametar optimizatora je korak za obučavanje (eng. learning rate) koji je u ovom radu na početku 0.05. Vrednost koraka za obučavanje je od ključne važnosti jer određuje brzinu konvergencije, odnosno kretanje ka optimalnim vrednostima težina mreže. Ukoliko je visok tada se brže teži optimumima ali može se desiti da se optimum preskoči. Niske vrednosti dovode do konvergencije ali su spore.

U model su uključene i funkcije povratnog poziva koje proveravaju kretanje vrednosti funkcije greške i tačnost za validacioni skup. Ukoliko se dogodi situacija da je došlo do platoa ili nekog lokalnog minimuma, poziva se funkcija povratnog poziva kako bi se te situacija prevazišle. Funkcija povratnog poziva koja je izabrana za smanjivanje koraka za obučavanje, u takvim situacijama, je eksponencijalna funkcija sa eksponentom  $-0.1$ , koji je uzet kao parametar prilikom pisanja koda i obučavanja modela.



## Glava 6

# Eksperimentalni rezultati

Model koji je opisan u glavi 5 je osnova digitalnog sagovornika otvorenog domena, što znači da može da generiše rečenice bez određene teme razgovora. Baza podataka koja je korišćena za potrebe ovog rada je pogodna za obučavanje digitalnih sagovornika koji su otvorenog domena upotrebe. Razlog korišćenja ove baze je jednostavnost strukture skupa podataka, ali i raznovrsnost podataka koji se u bazi čuvaju. Rezultat obučavanja je sagovornik sa kojim se komunicira putem terminala. Neki primeri razgovora se mogu videti u primerima 6.1 i 6.2. Primeri predstavljaju dva pokretanja sagovornika nakon obučavanja.

```
user: hello
chatbot: hello
user: what is your name
chatbot: jacob singer
user: how are you
chatbot: fine okay
user: are you a college student
chatbot: yes
user: do you have a job yet
chatbot: no
user: would you like a job at my restaurant
chatbot: i do not know
user: what is your college major
chatbot: victor frankenstein
user: do you like victor frankenstein
chatbot: yes
```

Primer 6.1: Prvi primer razgovora

```

user: hi
chatbot: hi
user: what is your name
chatbot: jacob singer
user: how are you
chatbot: i am fine i am fine
user: do you want some coffee
chatbot: no
user: i love you
chatbot: i love you too

```

Primer 6.2: Drugi primer razgovora

Dobijaju se rezultati koji su prikazani u tabeli 6.1. Razlog ovakve tačnosti je u parametrima koji se koriste prilikom obučavanja i skupovima rečenica koje se modelu prosleđuju. Pošto su skupovi relativno mali, sa rečenicama koje nisu mnogo dugačke postižu se naizgled loši rezultati.

Tabela 6.1: Prikaz tačnosti i gubitka

Skup podataka	Tačnost	Greška	BLEU mera
Skup za obučavanje	0.75	1.22	0.56
Skup za validaciju	0.68	2.68	0.55
Skup za testiranje	0.68	2.75	0.55

Međutim, mere koje se koriste prilikom evaluacije rečenice koje se generišu nisu savršene jer ne mere značenje i smisao generisane rečenice. Mera tačnosti izražava procenat tačno pogodjenih predviđanja u odnosu na ceo skup ali u kontekstu rečenica korišćenje ove mere nije dovoljna jer se rečenice formiraju nasumičnim izborom reči. To dovodi do razlikovanja nekoliko slučajeva odgovora i situacija u kojima bi se odgovor smatrao netačnim.

1. Ukoliko dobijena rečenica nije tačna: na primer, na pitanje *How are you?* se dobije odgovor *blue apple*.
2. Ukoliko dobijena rečenica nije tačna ali se smatra prihvatljivim odgovorom. Posmatrano iz ugla čoveka: na primer, na pitanje *How are you?* se dobije odgovor *I do not know*.
3. Ukoliko je rečenica tačna ali je odgovor drugačije iskazan: na primer, na pitanje *How are you?* se očekuje odgovor *i am fine* ali se dobio odgovor *fine*.

Opcije 2 i 3 se mogu smatrati prihvatljivim odgovorima jer se i prilikom razgovora sa čovekom očekuju takvi odgovori, dok je opcija 1 netačan odgovor. Ponašanje koje se može videti u primerima je očekivano zbog prirode skupa podataka koji se koristi za modelovanje digitalnog sagovornika [18] [6]. Zato se uvodi i dodatna mera, BLEU mera, koja prilikom izračunavanja vrednosti zahteva da se predviđene rečenice posmatraju kao jedinstvena celina.

```
user: do you like football
chatbot: yes
user: what is your favorite team
chatbot: american UNK
```

### Primer 6.3: Razgovor sa tokenom UNK

Sagovornik koji je nastao iz ovog modela generiše i rečenice koje nemaju smisla, kao i rečenice koje umesto neke konkretne reči sadrže token UNK, kao u primeru 6.3. Pojavljivanja tokena UNK je očekivano ponašanje jer je rečnik koji se koristi za obučavanje relativno mali i postoji dosta nepoznatih reči koje su zamenjene tim tokenom.

Dodatni razlozi dobijanja rezultata iz tabele 6.1 su ograničena memorija platforme, kao i sama struktura i organizacija baze podataka koje se koriste za obučavanje modela. Baza podataka koja se koristi u ovom radu je namenjena za kreiranje sagovornika sa otvorenim domenom i zasnovana je na dijalozima između likova koji u razgovoru učestvuju.

```
"I_hear_you're_helpin'_Verona."
"Uh, _yeah._We're_old_friend"
"You_and_Verona?"
"What?_We_took_bathes_together_when_we_were_kids."
```

### Primer 6.4: Primer dijaloga

Parovi rečenica pitanje–odgovor koji se koriste prilikom obučavanja nastaju tako što odgovor u jednom paru rečenica postaje pitanje u drugom paru rečenica. Ukoliko izdvojimo jedan dijalog kao u primeru 6.4, parovi se formiraju na način koji je dat u primeru 6.5.

```
I par
Pitanje: "I_hear_you're_helpin'_Verona."
Odgovor: "Uh, _yeah._We're_old_friend"
```

II par

Pitanje: "Uh, \_ \_ yeah. \_ \_ We' re\_ old\_ friend"

Odgovor: "You\_ and\_ Verona?"

III par

Pitanje: "You\_ and\_ Verona?"

Odgovor: "What? \_ \_ We\_ took\_ bathes\_ together\_ when\_ we\_ were\_ kids. "

Primer 6.5: Transformacija dijaloga u parove pitanje-odgovor

Ovakvim izdvajanjem dobija se raznolikost pitanja i odgovora koji se prosleđuju modelu prilikom obučavanja. Korišćenjem mehanizma pažnje kratki dijalozi se mogu lakše zapamtiti, ali očigledna mana ovog postupka je potencijalna nekompatibilnost rečenica koje nisu deo istog dijaloga. Nekompatibilne rečenice narušavaju poredak koji postoji unutar samih dijaloga jer predstavljaju loš prelaz između dijaloga i narušavaju tok razgovora.

## Glava 7

# Zaključak

Inteligentni sagovornici su veoma kompleksne aplikacije koje koriste procesiranje prirodnih jezika kao sastavni deo svoje funkcionalnosti i podrazumevaju veoma složenu arhitekturu. Model koji je pravljen za potrebe ovog rada je eksperimentalnog tipa i ima za cilj da ispita mogućnosti jednog modela neuronskih mreža. Rezultati koji su dobijeni su prihvatljivi i model može da generiše jednostavne rečenice na engleskom jeziku u skladu sa onim rečenicama koje je naučio.

Neuronske mreže kao sredstvo za učenje se poslednjih godina uspešno primenjuju u raznim oblastima. Prednosti koje pružaju su omogućile da postanu neophodan alat pri rešavanju kompleksnih problema. Za njihovu upotrebu potrebne su organizovane i velike baze podataka. Kako bi se napravio potpuno funkcionalni inteligentni digitalni sagovornik potrebno je dublje razmatranje pitanja „Šta znači biti inteligentan?”. Po definiciji psihologa, inteligencija je mentalna osobina koja se sastoji od sposobnosti: učenja iz iskustva, adaptiranje na situaciju, shvatanje i razumevanje novih situacija i korišćenje stečenog znanja. Slična definicija može da se primeni i na digitalne sagovornike i generalno na inteligentne mašine. Od jednostavnog alata do automatizovane zamene za ljudske delatnosti, inteligentne mašine su utemeljile ulogu neophodnog ljudskog pomagača. Međutim, inteligencija koju iskazuju ovakve mašine je veštački napravljena pod uticajem čoveka, odnosno programera. Da li će doći vreme kada će mašina moći sama da misli? Tehnologija svake godine sve više napreduje, sa novom tehnologijom nastaju nove mogućnosti i ideje koje nisu mogle da se ostvare prethodnom tehnologijom. U budućnosti možemo očekivati i digitalne sagovornike koji imaju dovoljno razvijen mehanizam pažnje da ga možemo smatrati dinamičkim, odnosno da ne zavisi od prvobitne konfiguracije programera. Uveliko se vrše eksperimenti u svrhu konstruisanja mašina sa drugim vrstama intelligen-

## *GLAVA 7. ZAKLJUČAK*

---

cije, kao što su emotivna inteligencija, muzička, interpersonalna pa i egzistencijalna. Savremeni eksperimenti digitalnih sagovornika sa emotivnom komponentom već postižu veoma dobre rezultate, a sa napretkom ideja i tehnologije možda i postanu neophodan deo svakodnevnog života.

# Bibliografija

- [1] Bayan AbuShawar and Eric Atwell. Alice chatbot: Trials and outputs. *Computación y Sistemas*, 19(4):625–632, 2015.
- [2] Eleni Adamopoulou and Lefteris Moussiades. An overview of chatbot technology. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 373–383. Springer, 2020.
- [3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016. <https://arxiv.org/pdf/1409.0473.pdf>.
- [4] Kenneth Mark Colby. Ten criticisms of parry. *SIGART Bull.*, (48):5–9, October 1974. url = <https://doi.org/10.1145/1045200.1045202>.
- [5] Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*, 2011.
- [6] Moein Hasani. Chatbot-with-tensorflow-and-keras. <https://github.com/Moeinh77/Chatbot-with-TensorFlow-and-Keras>, poslednji pristup: 2021-09-10.
- [7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.
- [8] Dara Milojković. Github stranica sa praktičnim delom rada, 2021. <https://github.com/Dara123M/Chatbot-Masters-Thesis>.
- [9] Mladen Nikolić and Anđelka Zečević. Mašinsko učenje, 2019. <http://ml.matf.bg.ac.rs/readings/ml.pdf>, poslednji pristup: 2021-09-06.

- [10] Stevan Ostrogonac. *Modeli srpskog jezika i njihova primena u govornim i jezičkim tehnologijama*. PhD thesis, Fakultet tehničkih nauka, Univerzitet u Novom Sadu, 12 2018. [https://www.cris.uns.ac.rs/DownloadFileServlet/Disertacija153796019798434.pdf?controlNumber=\(BISIS\)107812&fileName=153796019798434.pdf&id=12013&source=NaRDuS&language=sr](https://www.cris.uns.ac.rs/DownloadFileServlet/Disertacija153796019798434.pdf?controlNumber=(BISIS)107812&fileName=153796019798434.pdf&id=12013&source=NaRDuS&language=sr), poslednji pristup: 2021-08-12.
- [11] S. Pardeshi, S. Ovhal, P. Shinde, M. Bansode, and A. Birajda. A survey on different algorithms used in chatbot. *International Research Journal of Engineering and Technology (IRJET)*, 07:6092–6098, 2020.
- [12] Jagdish Singh, Minnu Helen Joesph, and Khurshid Begum Abdul Jabbar. Rule-based chabot for student enquiries. In *Journal of Physics: Conference Series*, volume 1228. IOP Publishing, 2019.
- [13] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014. <https://arxiv.org/pdf/1409.3215v3.pdf>.
- [14] Ange Tato and Roger Nkambou. Improving adam optimizer. 2018. <https://openreview.net/pdf?id=HJfpZq1DM>, poslednji pristup: 2021-08-14.
- [15] Richard Wallace. The elements of aiml style. *Alice AI Foundation*, 139, 2003.
- [16] J. Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9:36–45, 1966.
- [17] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The Design and Implementation of XiaoIce, an Empathetic Social Chatbot. *Computational Linguistics*, 46(1):53–93, 03 2020. [https://doi.org/10.1162/coli\\_a\\_00368](https://doi.org/10.1162/coli_a_00368).
- [18] Husein Zolkepli. Nlp-models-tensorflow. <https://github.com/huseinzol105/NLP-Models-Tensorflow/tree/master/chatbot>, poslednji pristup: 2021-09-10.