



Šolski center Celje
Srednja šola za kemijo, elektrotehniko in računalništvo

Sinteza govora

Raziskovalna naloga

Mentorja:

Valentina Hrastnik, prof.

mag. Boštjan Resinovič, univ. dipl. inž. rač. in inf.

Avtorja:

Nik Grebovšek, R4A

Marko Vrečer, R4A

Celje, junij 2020

Kazalo

1	Uvod	1
2	Hipoteze	2
3	Metode dela	2
4	Teoretični del	3
4.1	Kaj je sinteza govora?	3
4.1.1	Zgodovina	3
4.1.2	Vrste sintetizatorjev govora	4
4.2	Faze sinteze govora.....	5
4.2.1	Slovnična analiza.....	5
4.2.2	Grafemska pretvorba	5
4.2.3	Transkripcija	5
4.2.4	Določanje prozodičnih parametrov	5
4.2.5	Sinteza govora.....	5
5	Praktični del	6
5.1	MaryTTS	6
5.1.1	Osnovne zahteve.....	6
5.1.1.1	Operacijski sistem	6
5.1.1.2	Parametri vsakega SAMPA znaka	6
5.1.1.3	Slovar izgovorjav	6
5.1.1.4	Posnetki govora.....	6
5.1.2	Postopek izdelave govora	7
5.1.2.1	VoiceImportTools	9
5.1.2.1.1	Difonski sintetizator.....	9
5.1.2.1.2	PMM sintetizator	10
5.1.3	MaryTTS spletna stran	10
5.2	Globoko učenje (ang. deep learning)	11
5.2.1	Delovanje globoke nevronske mreže.....	11
5.2.2	Potrebna strojna in programska oprema	11
5.2.3	Sistem za učenje sintetizatorja govora.....	11
5.2.3.1	Priprava podatkovne zbirke	11
5.2.3.2	Uporaba podatkovne zbirke.....	12
5.2.3.3	Treniranje modela	12
6	Rezultati ankete	14

6.1	Starost.....	14
6.2	Ali ste že kdaj slišali za sintezo govora?.....	15
6.3	Poznate katerega izmed slovenskih sintetizatorjev govora?.....	15
6.4	Kako razumljiv se vam zdi 1. posnetek?.....	16
6.5	Kako razumljiv se vam zdi 2. posnetek?.....	16
6.6	Kako razumljiv se vam zdi 3. posnetek?.....	17
6.7	Kako razumljiv se vam zdi 4. posnetek?.....	17
6.8	Preizkus sintetizatorja	18
7	Razprava.....	19
7.1	1. hipoteza	19
7.2	2. hipoteza	19
7.3	3. hipoteza	19
8	Zaključek.....	20
9	Zahvala.....	20
10	Viri in literature	21
11	Priloge.....	22
11.1	Anketa.....	22
11.2	Izjava mentorice.....	24
11.3	Izjava mentorja	25

Kazalo slik

Slika 1:	Bell Labs VODER	3
Slika 2:	Speak & Spell	3
Slika 3:	Izdelava novega glasu - MaryTTS New Language support	7
Slika 4:	MaryTTS spletna stran.....	10
Slika 5:	Izgled grafa na začetku učenja (7500 korak).....	13
Slika 6:	Izgled grafa pri naučenem modelu (20000 korak).....	13
Slika 7:	Izgled Tacotronove spletne strani	13

Kazalo grafov

Graf 1:	Starost anketirancev	14
Graf 2:	Poznavanje sinteze govora	15
Graf 3:	Poznavanje slovenskih sintetizatorjev govora	15
Graf 4:	Ocene prvega glasu.....	16
Graf 5:	Ocene drugega glasu.....	16
Graf 6:	Ocene tretjega glasu.....	17
Graf 7:	Ocene četrtega glasu	17

Povzetek

V raziskovalni nalogi sva predstavila, kaj je sinteza govora, zgodovino sinteze in nekaj primerov možnosti sintetiziranja govora. Ob primerih sva se še posebej osredotočila na sintetizatorje govora slovenskega jezika. Poleg tega sva se lotila izdelave slovenskega sintetizatorja govora s pomočjo Prikritnega modela Markova (PMM) in globokega učenja.

Ključne besede: sintetizator slovenskega govora, PMM, globoko učenje, MaryTTS, Tacotron

Abstract

In this paper, we talked about what speech synthesis is, something about its history and some examples on how to make speech synthesis. Examples we gave are only for the Slovenian language. We also made a speech synthesis program for Slovenian language using Hidden Markov model (HMM) and deep learning.

Keywords: Slovenian speech synthesis, HMM, deep learning, MaryTTS, Tacotron

1 Uvod

Sporazumevanje je zelo zapleten postopek, s katerim tvorijo živa bitja medsebojne odnose. Če pogledamo že samo sporazumevanje med ljudmi, lahko opazimo več načinov nebesedne in besedne komunikacije. Poznamo različne jezikovne skupine, jezike in narečja, ki so si med seboj velikokrat zelo podobni. Če pogledamo načine sporazumevanja vseh živih bitij, lahko ugotovimo, da jih je še mnogo več. Ker je področje sporazumevanja tako široko, je tema začela zanimati vse več ljudi, ki se s tem področjem ukvarjajo profesionalno. Ti znanstveniki si prizadevajo načine sporazumevanja čim bolj razvozlati, s tem pa najti nove načine sinteze govora, ki bi mnogim olajšalo oz. poenostavilo življenje.

2 Hipoteze

Pred začetkom in med izdelovanjem naloge sva si postavila nekaj hipotez:

- Z enim izmed uporabljenih orodij nama bo uspelo izdelati slovenski sintetizator govora.
- Z globokim učenjem (ang. deep learning) bova dobila bolj razumljiv glas kot z drugimi metodami.
- Končni glas bo razumljiv.

3 Metode dela

Pri izdelovanju raziskovalne naloge sva si pomagala z različnimi metodami dela:

- raziskovanje in branje člankov o sintezi govora,
- snemanje glasu za MaryTTS,
- obdelava zvočnih posnetkov za globoko učenje,
- anketiranje dijakov,
- primerjava in vrednotenje pridobljenih rezultatov.

4 Teoretični del

4.1 Kaj je sinteza govora?

Sinteza govora je postopek pretvorbe nekega vhodnega besedila v človeku razumljiv glas.

4.1.1 Zgodovina

Zgodovina sinteze govora človeškega jezika sega že v prvo polovico 20. stoletja. Prva naprava, ki je uspešno izvajala sintezo govora, imenovana VODER, je bila izdelana leta 1939 v podjetju Bell Labs. Ta naprava ni samodejno pretvarjala vhodnega besedila v sintetiziran glas, ampak jo je morala upravljati oseba. Imela je petnajst tipk, s katerimi je upravljalec izdeloval glasove, prilagajal frekvenco, hitrost, glasnost itd. Glas je bil izključno elektronski in ni bil najbolj prijeten za poslušanje.

Leta 1962 je isto podjetje napisalo revolucionarni program za IBM-ov računalnik 704, ki ni omogočil samo to, da je lahko govoril, temveč tudi to, da je računalnik pel. Glas je bil podobno kot prej le elektronski.



Slika 1: Bell Labs VODER

Prvo podjetje, katerega glavni namen je bil, da se ukvarja s sintezo govora, je bilo Votrax, ki ga je ustanovil Richard Gagnon leta 1970. Podjetje je izdelovalo razširitvene kartice in zunanje module za uporabnike, ki so potrebovali računalnik s sposobnostjo govora.



Slika 2: Speak & Spell

Leta 1978 je podjetje Texas Instruments izdelalo igračo Speak & Spell, ki je pomagala otrokom, da so se naučili črkovati angleške besede. Ker je podjetje želelo narediti čim cenejšo igračo, so se odločili, da bo glasove izdelovala sproti, zato ne bi bilo treba posneti vseh besed jezika. Narejena je bila tako, da je združevala foneme, s katerimi je bilo mogoče "zlepiti" razumljive besede. Texas Instruments je po izdelavi Speak & Spell-a izdal tudi računalniški modul za njihov osebni računalnik TI-99, s katerim so lahko programerji uporabili glas v svojih aplikacijah, kar je bilo za tisti čas precej napredno. Glas je bil razumljiv, vendar za današnje standarde preveč robotski.

Eden izmed prvih programov, ki za sintezo ni potreboval nobenih zunanjih modulov, je bil S. A. M. (Software Automatic Mouth), prvič izdan leta 1979. Prilagojen je bil za kar nekaj operacijskih sistemov, med katere spadajo Apple-ovi sistemi, Atari osebni računalniki in Commodore 64. Ker za izdelavo glasu ni potreboval dodatne strojne opreme, je sinteza trajala nekoliko dlje.

Po letu 1980 so raziskovalci začeli uporabljati nove in drugačne pristope za izdelovanje sintetizatorjev govora. Še vedno so poizkušali s sintetizatorji, ki »lepijo« difone/besede/zloge skupaj, vendar so ugotovili, da za naravnejšo sintezo potrebujejo nove pristope. Ugotovili so, da lahko uporabijo Prikriti model Markova, ki se v matematiki in statistiki uporablja za predvidevanje spremenljivk na podlagi nekkih podatkov, v našem primeru na podatkih s posnetkov. Eno izmed znanih podjetij Microsoft je v svoje

produkte dodalo možnost sinteze šele leta 1995, vendar so na trgu obstajala druga podjetja, ki so se s tem še posebej ukvarjala.

Prvi sintetizatorji govora za slovenščino so se začeli pojavljati šele leta 1997. Prvi sintetizator, imenovan S5, je bil izdelan na Fakulteti za elektrotehniko v Ljubljani za operacijski sistem MS-DOS. Pri tem je sodelovala gospa Jerneja Gros, ki še danes sodeluje pri izdelovanju slovenskih sintetizatorjev. Prvi javni produkt, ki ga je bilo mogoče kupiti na slovenskem trgu, je bil Amebisov Govorec, izdan leta 2006. Prav tako je bil naslednje leto izdan Alpineonov Proteus, a ni bil namenjen širokemu trgu.

Naslednji sintetizator za slovenski jezik, ki je bil precej pomemben, je bil izdan šele leta 2015 pod imenom eBralec in je danes najboljši sintetizator za slovenski jezik. Izdali so ga Institut Jožef Stefan, Amebis in Alpineon.

Najnovejši slovenski sintetizator, izdan leta 2017, je Microsoft Lado.

4.1.2 Vrste sintetizatorjev govora

Sinteza govora lahko poteka na več različnih načinov:

- Združevanje delov enot.
 - Enotsko združevanje združuje določene enote v govor. Enote so lahko deli glasu, glasovi, zlogi, besede, besedne zveze, stavki. Za takšne sintetizatorje potrebujemo zelo velike količine posnetkov, če želimo, da je govor čim bolj razumljiv. Kot primer lahko vzamemo enoto besedo. V slovenščini ima lahko ena beseda veliko različnih oblik. Na primer en samostalniik ima najmanj šest različnih oblik, kar bi pomenilo zelo velike količine podatkov. Težava bi lahko nastala pri homografih, saj se lahko zgodi, da sintetizator vzame napačno besedo, ker ne razume pomena stavka. Primer: **Gori na gori gori**.
 - Difonska sinteza temelji na lepljenju difonov, s katerimi lahko nastane razumljiv glas. Velikost podatkovne zbirke v tem primeru ni tako velika, saj mora vsebovati samo vse možne difone, namesto celotnih besed. Slovenski sintetizator eBralec, ima v svoji zbirki 1883 difonov.
 - Posebno združevanje se uporablja predvsem tam, kjer si je sintetizirano besedilo zelo podobno. Takšni sistemi se uporabljajo predvsem v prevozništvu. Primer je avtobusna postaja, kjer avtomatski sistem obvesti vozače o voznem redu.
- Formantni sintetizatorji posnemajo človeški glas z nastavljanjem amplitude, frekvence in ostalih parametrov. Takšne načine sinteze hitro prepoznamo, da niso človeški, saj imajo značilen "robotski" glas. Uporablja se predvsem tam, kjer kvaliteta glasu ni tako zelo pomembna. Ta način je bil uporabljen tudi v igračih Speak & spell.
- Artikulatorni sintetizatorji posnemajo govorno cev. To pomeni, da simulirajo človeške organe, ki se uporabijo za proizvodnjo zvoka. Ta način se ne uporablja v komercialnih produktih predvsem zaradi kompleksnosti.
- PMM/HMM sintetizatorji – Prikriti model Markova/Hidden model Markova je način, ki uporablja statistiko in sklepanje za proizvodnjo govora.
- T. i. globoko učenje (ang. deep learning) se je začelo popularizirati šele v zadnjih nekaj letih z napredkom strojne opreme, primerne za učenje. Ta metoda se uči glede na že posnet govor ter ga s pomočjo umetne inteligence izboljšuje. Najbolj popularno orodje je Tacotron, ki je javno dostopen in odprtokoden.

4.2 Faze sinteze govora

Sinteza govora poteka v določenih fazah. Pri nekaterih sintetizatorjih so lahko faze različne, izpuščene, dodane itd. Osredotočila sva se predvsem na faze sinteze pri slovenskem sintetizatorju eBralec.

4.2.1 Slovnična analiza

Na začetku se izvede slovnična analiza vhodnega besedila. eBralec in Amebisov Govorec najprej določita vsaki besedi besedno vrsto, število, sklon, spol idr. iz slovarja izgovorjav. Te nastavitve so pomembne, da sintetizator ve, kaj je prebrana beseda. Slovnična analiza je enostavnejša za ljudi, saj lahko na podlagi izkušenj hitro sklepamo in razmislimo, kaj bi beseda lahko bila, računalnik pa se mora zanašati na slovar izgovorjav, v katerem so podatki o besedi.

V tej fazi se lahko zgodijo napake pri analizi, ki so odgovorne za to, da so besede nepravilno naglašene. Konkreten primer je »Gori na gori gori«. Ljudje ta stavek preberemo brez težav in vemo, kaj pomeni. Računalnik pa vidi tri besede, ki so enake. Govorec tega stavka ne izgovori pravilno, medtem ko ga eBralec zaradi izboljšanja slovarja lahko.

4.2.2 Grafemska pretvorba

Grafemska pretvorba pretvori vhodno besedilo v besede. V vhodnem besedilu zamenja vse simbole, številke, okrajšave itd. v besede, da se jih lahko v naslednji fazi pretvori naprej. V tem delu mora prepoznati ločila kot skladijska ali neskladijska, ali je pika zaključila stavek ali okrajšavo.

Po končanem postopku je vhodno besedilo pretvorjeno v grafemski zapis, ki je pripravljena za transkripcijo.

4.2.3 Transkripcija

Je faza, ki pretvori grafemski zapis v fonetičnega. To pomeni, da vsako besedo poišče v slovarju izgovorjav, kjer najde zapis o tem, kako se ta izgovori. Če besede v slovarju ne najde, jo zapiše z osnovnimi simboli.

Transkripcijo lahko zapisujemo na več načinov:

- IPA (International phonetics alphabet) – posebni znaki (npr. ʃ, θ, ʈ, ɔ, ε, ə ...),
- SAMPA (Speech Assessment Methods Phonetic Alphabet) – uporaba črk za označevanje (npr. A, @, }, B, V, Y ...),
- MRPA (Machine readable phonetics alphabet) – izpeljanka SAMPA-e,
- ...

Za slovar izgovorjav sva uporabila slovar SloLeks 2.0, ki ima besede zapisane v SAMPA obliki, kar nama je olajšalo delo pri praktičnem delu.

4.2.4 Določanje prozodičnih parametrov

Med prozodične lastnosti štejemo naglaševanje, trajanje, višino in grupiranje besed. Ti parametri so zelo pomembni za pravilno izgovorjavo besed, saj bi bile brez njih te nerazumljive.

4.2.5 Sinteza govora

Zadnji korak je sinteza govora. Ta del prevzame transkriptirano besedilo, vključno s prozodičnimi parametri in na podlagi vseh teh podatkov izdelava zvok oz. glas.

5 Praktični del

Odločila sva se, da bova poskusila narediti sintetizator na dva načina. Uporabila sva programski paket MaryTTS in Tacotron s TensorFlow.

5.1 MaryTTS

MaryTTS je odprtokodni programski paket namenjen izdelovanju sintetizatorjev govora na podlagi PMM ali difonske sinteze. Razvijati so ga začeli na nemškem raziskovalnem centru za umetno inteligenco (German Research Center for Artificial Intelligence). Napisan je v Javi, koda pa je prosto dostopna na spletni strani GitHub.

5.1.1 Osnovne zahteve

5.1.1.1 Operacijski sistem

Večina orodij za sintetiziranje govora je primarno narejena za poganjanje v Linux okolju. Zaradi tega sva se odločila, da bova na začetku uporabila WSL Ubuntu (Windows Subsystem for Linux), kasneje, če bo potrebno, pa bi uporabila celotno namestitvev Ubuntu.

WSL je program, ki nam omogoča zaganjati programe, narejene za Linux na operacijskem sistemu Windows. Za ta program sva se najprej odločila, saj je dandanes del vsake nove namestitve Windows 10 in sva sklepala, da bo zadostil zahtevam.

5.1.1.2 Parametri vsakega SAMPA znaka

Pri zbiranju parametrov za SAMPA znake sva izgubila precej časa, saj podatki za slovenščino niso prosto dostopni. Odločila sva se, da bova uporabila parametre za hrvaški jezik, ki sva jih nato dodatno spremenila. Vsi SAMPA podatki so zbrani v XML formatu.

5.1.1.3 Slovar izgovorjav

MaryTTS pridobi besede za učenje s slovenske Wikipedije in jih shrani v posebno datoteko. Najina naloga bi bila transkriptirati čim več teh besed. Ker pa je bilo besed preveliko (več tisoč), se nama je to zdel prevelik zalogaj, zato sva začela raziskovati, kje bi bili podobni podatki na voljo. Kontaktirala sva Roberta Rozmana, predavatelja s Fakultete za računalništvo in informatiko v Ljubljani, ki je povprašal na Amebisu, kje bi se lahko ti podatki dobili. Predlagali so projekt SloLeks 2.0, ki vsebuje besede napisane tudi v SAMPA obliki, kar nama je precej olajšalo delo.

Tabela 1: Primer besed v SAMPA obliki

fotoaparatu	fo:tOapara:tu
muzikologi	muzikOlo:gi
betonska	bEto:naska
naprtili	nap@rtili
skušaš	sku:SaS
črvivo	tSrvivO

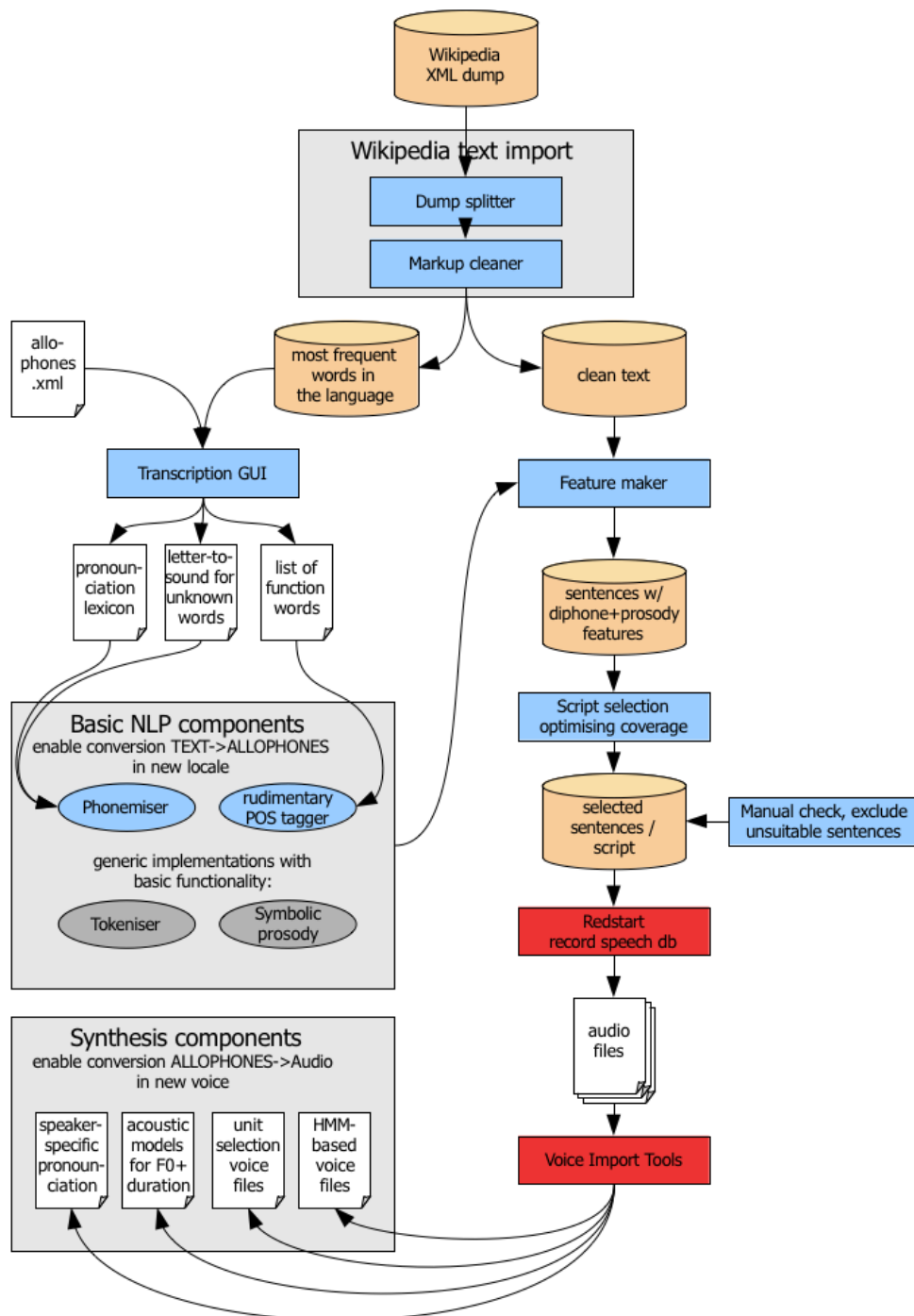
5.1.1.4 Posnetki govora

Posnetke govora sva naredila sama s pomočjo orodja RedStart; besedilo, ki sva ga prebrala, je bilo pridobljeno z Wikipedije.

5.1.2 Postopek izdelave govora

Delo, povezano z MaryTTS, sva opravljala na operacijskem sistemu Windows 10 z WSL in na Ubuntu.

Postopek izdelave govora je prikazan v spodnji sliki. Izdelava glasu je razdeljena na dva dela. V prvem delu pripravimo podatke za nov jezik, za katerega želimo izdelati glas, v drugem pa izberemo, kateri način sinteze bomo uporabili.



Slika 3: Izdelava novega glasu - MaryTTS New Language support

MaryTTS za pravilno delovanje in izdelavo leksikona potrebuje podatkovno bazo. Za to sva uporabila programski paket XAMPP s podatkovno bazo MySQL in Apache spletnim strežnikom.

MaryTTS za izdelovanje novega jezika uporablja Wikipedijino podatkovno bazo člankov. Program razdrobi najnovejše članke na manjše dele in izdela seznam vseh besed ter jih razvrsti padajoče po pogostosti.

Naslednji korak je bila transkripcija teh besed. Za to sva uporabila slovar SloLeks 2.0 (glej poglavje 4.1.1.3). Napisala sva C# program, ki najprej iz XML datoteke SloLeks-a pretvori v SAMPA zapis, tako da je v končni datoteki v eni vrsti beseda, zraven pa njen zapis. Ta postopek traja precej dolgo. Program naredi datoteko izvoz.txt, ki jo lahko uporabljamo naprej.

Kasneje sva ugotovila, da SAMPA zapis vsebuje narekovaje za označevanje naglasa in sva jih morala odstraniti, saj MaryTTS tega ne zna uporabljati. Prav tako sva ugotovila, da so nekatere črke zapisane v nenavadni obliki, zato sva dodala podprogram, ki izbriše narekovaje in posebne črke ter naredi novo datoteko izvoz_nov.txt.

Naslednji korak, ki je trajal približno 18 ur, je bil iskanje ujemanja med slovarjem najpogostejših besed iz Wikipedije in slovarjem SloLeks-a. Za to sva napisala podprogram, ki išče ujemanja med tema dvema slovarjema in sproti shranjuje besede, za katere najde par in za katere ne. Ta program naredi datoteko potrebno_dodati.txt in popolni_rezultati.txt, ki jo lahko uporabljamo naprej.

Zadnjo datoteko uvozimo v program transcription, ta naredi datoteke, ki se bodo uporabile pri izdelavi glasu. Te datoteke so pomembne pri pretvorbi besedo v glasove, s katerimi bo program vedel, kako besedo pravilno izgovoriti in naglasiti.

V naslednjem koraku izdelamo del programa, t. i. NLP (Natural Language Processing) komponento, katere namen je, da dobljeno besedilo pretvori v takšno obliko, da ga sintetizator lahko prebere. Izbrala sva NLP komponento za ruščino in jo spremenila tako, da ustreza slovenskemu jeziku.

Naslednji korak s pomočjo NLP komponente po Wikipedijinih člankih razvrsti vse stavke kot zanesljive in nezanesljive ter jih shranjuje v podatkovno bazo. Ta del označi stavke kot nezanesljive, če imajo znake, številke in dele, ki so programu neznani. Ta del traja približno 8 ur.

V naslednjem koraku z grafično aplikacijo izberemo stavke, ki se nam zdijo v redu in zanesljivi za branje in se lahko uporabijo naprej. Midva sva jih najprej izbrala 90, kasneje pa sva se odločila, da bova posnela novih 180. Odstranila sva stavke, ki vsebujejo rimske številke, številke, nekatere angleške besede (imena, kraji) in stavke s slovničnimi napakami.

Sledilo je snemanje teh stavkov. MaryTTS ima v ta namen zelo priročno orodje RedStart. Program zahteva, da je vsak stavek v ločeni datoteki. Ker tega prejšnji program ni naredil, sva napisala C# program, ki vsak stavek, izbran v prejšnjem koraku, shrani v novo datoteko s številko, ki se povečuje. RedStart omogoča tudi hitrejšo snemanje, saj sam ugotovi, kdaj smo prebrali stavek, ustavi snemanje in tako lahko nadaljujemo z drugim posnetkom.

Po snemanju posnetkih je sledila njihova pretvorba v drugačen format. MaryTTS zahteva posebne posnetke s specifikacijami:

- posnetek v mono obliki in
- sample rate 16 kHz.

Vključen program ponudi možnost, da obdelava le zadnji - najboljši posnetek vsakega stavka in odstrani tišino in zvoke, ki jih človek ustvarja med govorom.

Ti postopki so za PMM in lepljenje difonov enaki. Ko naredimo te korake, lahko nadaljujemo z izdelavo glasu.

5.1.2.1 *VoiceImportTools*

VoiceImportTools je program za izdelovanje glasu. Z njim lahko izdelamo difonski sintetizator ali PMM sintetizator. Najprej sva se lotila difonskega sintetizatorja.

5.1.2.1.1 Difonski sintetizator

Difonski sintetizator iz posnetkov najprej pridobi akustične parametre in intonacije fonemov.

V naslednjem koraku s pomočjo HTK (Hidden Markov Model Toolkit) pregleda vse posnetke in informacije iz prejšnjega koraka. Ta korak najde foneme v posnetku in jih poravnava. V zadnjem delu tega koraka lahko z oknom brskamo po teh fonemih in preverimo, ali so pravilno poravnani.

Iz posnetkov pridobi vektorske podatke fonemov in kasneje preveri, ali so vse datoteke na pravem mestu in če ima program vse, kar potrebuje.

V naslednjem koraku se združi vse posnetke v eno datoteko, prav tako pa naredi neke vrste časovnico posnetkov. Poleg tega se pripravi še dodatne parametre in datoteke za končni korak, ki vse to združi v glas.

V zadnjem koraku se zgradi glas iz vseh podatkov, da ga lahko uporabimo.

5.1.2.1.2 PMM sintetizator

V prvem koraku pripravimo podatke za treniranje PMM glasu. S pomočjo alofonov označi glasove v posnetkih z orodjem FestVox. Proces je v najinem primeru trajal skoraj cel dan, saj sva program poganjala na nekoliko starejšem prenosnem računalniku.

Ta program odstrani tišino in izdela datoteke, uporabljene kasneje pri treniranju novega glasu.

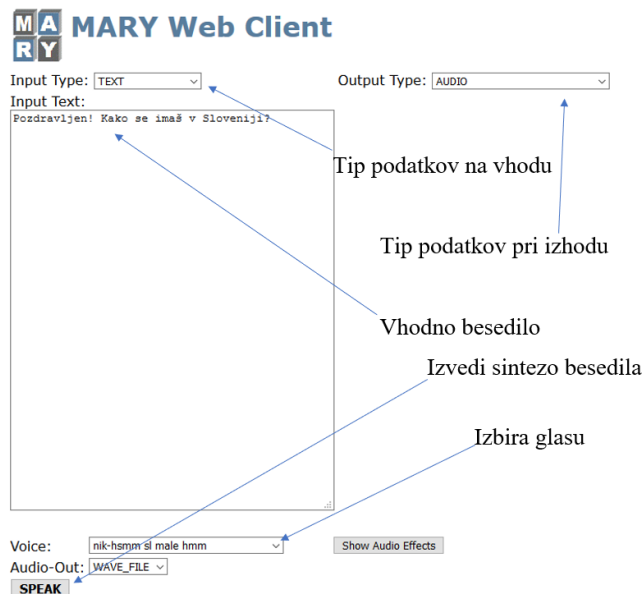
Treniranje novega glasu traja približno 2 uri. Ta del lahko ponovimo večkrat, da dosežemo boljšo kvaliteto glasu. Pri tem sva imela kar nekaj težav s programskim jezikom Perl, saj starejše verzije, ki jo MaryTTS zahteva, ni bilo več na voljo za prenos. Iskanje prave verzije je trajalo nekaj ur, vendar sva jo na koncu našla na eni zmed starejših spletnih strani, namenjeni PMM sintetizatorjem.

V zadnjem koraku VoiceImportTools zgradi nov glas s pomočjo programskega paketa Maven.

Sedaj glas uvozimo v MaryTTS strežniški program, da ga bomo lahko kasneje uporabili na spletni strani.

5.1.3 MaryTTS spletna stran

Za uporabo glasu in ostalih komponent ima MaryTTS vgrajen spletni strežnik, s katerim lahko upravljamo preko brskalnika. Če ga zaženemo, lahko na lokalnem računalniku dostopamo preko <http://localhost:59125>. Na spletni strani lahko izberemo vhodne in izhodne podatke. Najpogosteje uporabljena nastavitvev je že nastavljena. Vhod je navadno besedilo, izhod pa zvok, natančneje wav datoteka.



Slika 4: MaryTTS spletna stran

5.2 Globoko učenje (ang. deep learning)

Globoko učenje je postopek, ki temelji na umetni inteligenci in omogoča neki programski opremi, da se uči na način, podoben človeškemu možganom. Takšno mrežo sestavljajo posebne enote, t. i. tenzorji (podobno kot nevroni v naših možganih), ki se na podlagi predznanja in učenja povezujejo v nevronske mreže. Ko ima ta nevronska mreža več slojev, lahko govorimo o globokem učenju.

Zaradi takšnega načina pomnjenja podatkov lahko sistemi, ki se učijo s postopkom globokega učenja, celo presegaajo možgansko zmogljivost ljudi, kar je bilo še pred nekaj leti nepredstavljivo.

5.2.1 Delovanje globoke nevronske mreže

Sistem, naučen z globokim učenjem, deluje tako, da ko dobi nek vhodni podatek (npr. sliko psa), na vsakem tenzorju določi vrednost, ki predstavlja, kako verjetno je, da je na sliki iskana žival. Tako imamo lahko na prvem sloju tenzorjev okvirne podatke (npr. kakšne oblike je, ali ima noge itd.), na višjih oz. bolj specializiranih pa podrobne podatke (npr. velikost gobca, oblika oči itd.).

Kategorije tenzorjev si določi algoritem sam v postopku učenja, prav tako pa se lahko s povratnimi informacijami uporabnika sam izboljšuje.

5.2.2 Potrebna strojna in programska oprema

Za globoko učenje je eden izmed najbolj znanih programov Googlov TensorFlow, ki je na voljo od leta 2017. Primarno je napisan v programskih jezikih Python in C++ in omogoča učenje tako na procesorju kot tudi na grafični kartici. Slednji način je hitrejši, saj TensorFlow podpira uporabo CUDA jeder. To so posebna jedra, ki omogočajo hitrejšo izvajanje matematičnih operacij (podobno kot bi imeli več CPE na grafični kartici). Takšna jedra lahko najdemo na nekaterih nVidijinih grafičnih karticah, ki so namenjene predvsem profesionalni uporabi.

Za učenje najine nevronske mreže sva uporabila Python 3.6.8 in TensorFlow 1.14.0, saj sva ugotovila, da ti verziji povzročata najmanj problemov pri poganjanju projekta, ki sva ga uporabila. Koda je sicer napisana za Linux, vendar sva jo prilagodila tudi za delovanje na Windowsu.

Za globoko učenje seveda potrebujemo čim zmogljivejši računalnik, da se bo celotni proces učenja zaključil čim hitreje. Zaradi tega nama je bil preko šole omogočen dostop do Arcturjevega superračunalnika, na katerem sva lahko naučila nevronske mreže, za testiranje pa sva uporabila tudi osebni računalnik.

5.2.3 Sistem za učenje sintetizatorja govora

Kot je bilo omenjeno v prejšnji točki, sva za učenje najine nevronske mreže uporabila projekt Tacotron. Projekt je odprtokoden in prosto dostopen na GitHub-u in je eden izmed najpopularnejših programov za proizvodnjo sinteze govora s pomočjo globokega učenja.

5.2.3.1 Priprava podatkovne zbirke

Za to, da lahko sploh začnemo učiti nevronske mreže, moramo najprej pridobiti podatke. Te bi lahko snemala sama, vendar sva se odločila, da bi bilo bolje, da bi bili posnetki narejeni profesionalno. Tako sva zasledila, da je pred kratkim izšla na RTV zvočna knjiga Jezero, avtorja Tadeja Goloba, ki jo bere Sebastian Cavazza. Z urednico Gabrijelo Gruden smo se dogovorili, da jo lahko uporabimo za izključno šolske namene. Poleg branja sva pridobila tudi besedilo, da bo lahko program z njim primerjal izgovorjavo in se tako učil.

Pri izdelavi uporabnih podatkov sva se zgledovala po podatkovni zbirki LJSpeech, ki je prosto dostopna, vendar v angleščini. Posnetki, ki jih zajema, so dolgi približno 24 ur, posamezno pa od 1 do 10 sekund.

Najini pridobljeni posnetki so bili sprva dolgi od 15 do 20 minut, kar je precej predolgo za smiselno učenje. Zato sva podatke najprej avtomatično razdelila s programom WavePad Sound editor glede na tišine, ki so se pojavile v besedilu. Izkazalo se je, da so tudi takšni posnetki predolgi za učenje, saj so v večini zajemali več stavkov, kar pa otežuje delo algoritmu za globoko učenje. Da bi algoritmu olajšala delo, sva se odločila, da bova predolge posnetke ročno razdelila v krajše enote.

Pri tem sva si pomagala s programom Audacity, ker omogoča dokaj enostavno izvažanje posamičnih delov govora. Tako sva na koncu dobila 1104 glasovnih datotek s skupno dolžino 104 minute, za pripravo katerih sva potrebovala okoli 7 ur.

Kljub temu da je to precej manj kot pri podatkovni zbirki LJSpeech, sva mnenja, da bodo podatki zadostovali.

Ko sva končala z »rezanjem« posnetkov, sva jim morala dodati besedilo in ga oštevilčiti. Naredila sva seznam predvajanja posameznega poglavja in ob spremembi posnetka v besedilu naredila novo vrsto.

To besedilo sva nato vstavila v C# program, ki je avtomatično oštevilčil vsako vrstico v besedilu glede na ime datoteke.

Postopek dodajanja besedila k »narezanemu« govoru nama je vzel približno 5 ur.

Celotni postopek ustvarjanja podatkovne zbirke sva zaključila tako, da sva vse posnetke združila v eno mapo, dodala datoteko z metapodatki (kjer je oštevilčeno besedilo glede na posnetek), ki vsebuje številko posnetka, nato napisano in še govorjeno besedilo (v najinem primeru je oboje enako).

5.2.3.2 *Uporaba podatkovne zbirke*

Da lahko podatkovno zbirko uporabimo s Tacotromom, jo moramo najprej »registrirati« in vnaprej obdelati.

Da novo podatkovno zbirko »registriramo«, dodamo novo datoteko v mapi »datasets«, kjer se določi podatkovna struktura zbirke. Midva sva se odločila, da bo najina struktura enaka kot pri LJSpeech, zato nisva imela veliko težav z določanjem parametrov, saj so bili ti večinoma enaki oz. podobni.

Vnaprejšnja obdelava se naredi s pomočjo programa »preprocess.py«, ki ustvari datoteke podatkovne zbirke s končnico .npy (Numpy), v katero združi hkrati podatke o besedilu in govoru. Ta format datoteke uporabi zato, ker je dostop do podatkov v njej bistveno hitrejši kot iskanje posameznega vnosa po metapodatkih, nato pa še iskanje in obdelovanje zvočnega posnetka.

Preden končno začnemo model trenirati, se lahko odločimo spremeniti še nekatere parametre treniranja (npr. koliko ponovitev želimo, koliko podatkov naenkrat obdelujemo, kako pogosto se shranjujejo datoteke itd.).

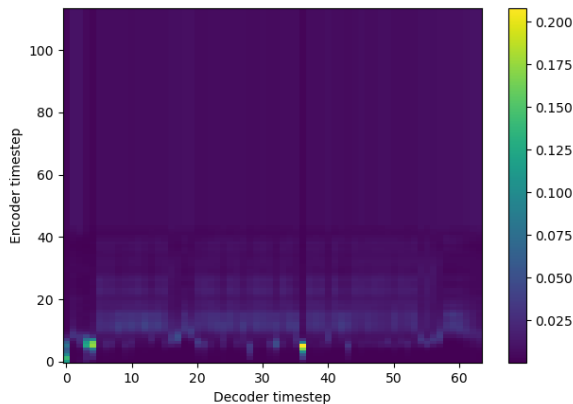
5.2.3.3 *Treniranje modela*

Ko smo končali s predpripravo podatkov, lahko začnemo s treniranjem modela s programom »train.py«.

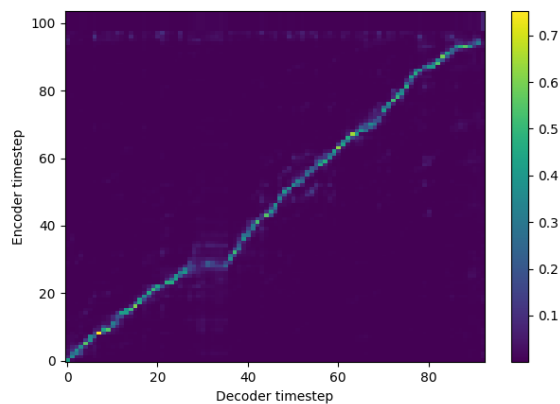
Vsakič ko se naredi nov korak (ang. step), nas program o tem obvesti, prav tako pa nas obvesti o trenutni statistiki modela. Iz nje lahko izvemo njegovo uspešnost pri učenju in kako hitro se model uči. Hitrosti učenja so odvisne od velikosti vhodnih podatkov in od strojne opreme računalnika. Najin model je na

osebni računalnik prek CPE naredil en korak na približno 7 sekund, medtem ko je na GPE naredil korak na približno eno sekundo in pol. Na superračunalniku so bile številke precej podobne, verjetno zaradi omejitve zmogljivosti programske opreme – Pythona.

Program torej sam na določeno število korakov shrani svoj napredek. Najini parametri so določali, da se shrani na vsakih 250 korakov. Poleg datoteke svojega napredka (»checkpoint«) shrani tudi posnetek izgovorjave trenutnega stavka in sliko, ki prikazuje graf uspešnosti delovanja pretvarjanja besedila v govor. Da bi bil postopek pretvarjanja čim uspešnejši, mora biti graf čim bolj linearen (glej Sliko 6) – torej mora biti razmerje med kodiranjem in dekodiranjem čim enakomernejše.

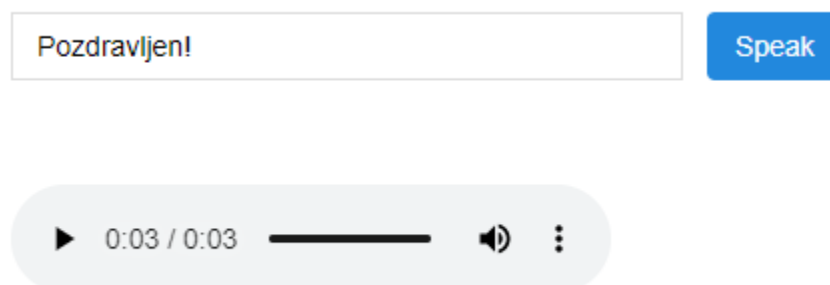


tacotron, None, 2020-03-07 20:22, step=7500, loss=0.09795
Slika 5: Izgled grafa na začetku učenja (7500 korak)



tacotron, None, 2017-07-06 06:14, step=20000, loss=0.09747
Slika 6: Izgled grafa pri naučenem modelu (20000 korak)

Ko smo zadovoljni s treniranjem, lahko testiramo uspešnost našega modela s programom »demo_server.py«. V program moramo kot parameter vnesti tudi model, ki ga želimo uporabiti (navedemo »checkpoint« datoteko). Program zažene spletno stran, na kateri lahko vnašamo besedilo, ki ga želimo sintetizirati.



Slika 7: Izgled Tacotronove spletne strani

6 Rezultati ankete

V anketi sva se osredotočila predvsem na to, kateri izmed glasov se uporabnikom zdi najbolj uporaben. Anketirancem med reševanjem ankete nisva podala informacij o tem, kateri izmed glasov je za izdelavo uporabljal 175, 200 ali 400 posnetkov in s kakšno metodo je bil narejen, da bi bili rezultati čim bolj objektivni.

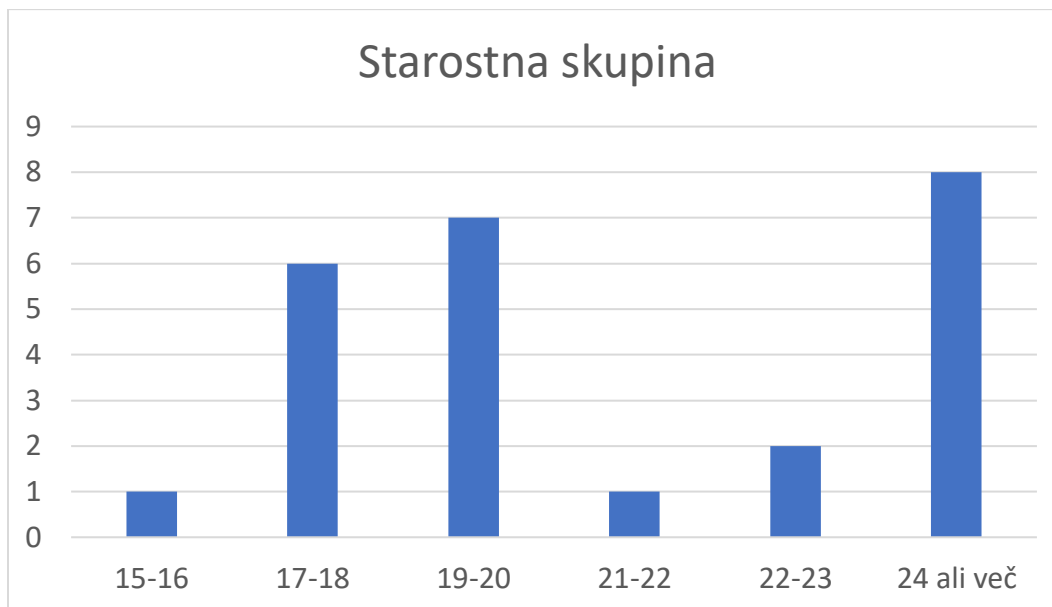
Anketo se reševali dijaki 3. in 4. letnika Srednje šole za kemijo elektrotehniko in računalništvo ter člani Facebook skupin Matura 2020, Slovenski programerji in Slovenski junior programerji. Rešilo jo je 25 anketirancev. Anketo so reševali preko Microsoft Office Forms obrazca. Pri vseh štirih glasovih je glas prebral:

»Na Srednji šoli za kemijo, elektrotehniko in računalništvo se zavedamo tega dejstva v kontekstu družbenih razmer. Z dolgoletno tradicijo skrbimo za uspešen kader v gospodarstvu, predvsem pa spodbujamo dijake k nadaljnjemu strokovnemu sodelovanju ter izpopolnjevanju, s čimer prispevamo k njihovi osebni rasti, kar se posledično odraža tudi v širši družbi.

Zgodovina Šolskega centra Celje sega v leto tisoč devetsto devetinpetdeset, izobraževanje na področju kemije poteka na šoli od leta tisoč devetsto šestinsedemdeset, na področju elektrotehniko od leta tisoč devetsto osemindemdeset, zadnji program s področja računalništva je bil vpeljan leta dva tisoč dva.«

6.1 Starost

Prvo vprašanje je bilo demografskega tipa, v katerem sva anketirance vprašala po starosti. Rezultati tu so pričakovani, glede na skupine v katere sva anketo oddala, in sicer je bilo največ anketirancev starejših od 24 let, nato pa so sledili srednješolci četrtega in tretjega letnika.



Graf 1: Starost anketirancev

6.2 Ali ste že kdaj slišali za sintezo govora?

V drugem vprašanju sva anketirance vprašala, če so že kdaj slišali za sintezo govora. Na to vprašanje so pritrdilno odgovorili pričakovano s 60 %. Le 40 % je odgovorilo, da zanjo še niso slišali.



Graf 2: Poznavanje sinteze govora

6.3 Poznate katerega izmed slovenskih sintetizatorjev govora?

V tretjem vprašanju sva anketirance vprašala, če poznajo katerega izmed najbolj znanih slovenskih sintetizatorjev. Pričakovano je 44 % odgovorilo, da poznajo program eBralec, 25 % jih pozna sintetizator Microsoft Lado, program Proteus pa je poznalo le 19 %.



Graf 3: Poznavanje slovenskih sintetizatorjev govora

6.4 Kako razumljiv se vam zdi 1. posnetek?

S četrtem vprašanjem sva povprašala po oceni, kakšen se anketirancem zdi glas, narejen s 175 posnetki in PMM metodo. Povprečna ocena je 2,72. Rezultati so pričakovani, saj je PMM eden izmed najpogosteje uporabljenih načinov sinteze.

Kot razlog, zakaj so ga tako ocenili, so povedali, da se glas sliši precej robotsko, s čudnim naglaševanjem besed in da včasih glas prekinja.



Graf 4: Ocene prvega glasu

6.5 Kako razumljiv se vam zdi 2. posnetek?

S petim vprašanjem sva povprašala po oceni glasu, ki je bil narejen s pomočjo unit selection-a in 400 posnetki (verzija 2.0). Povprečna ocena je 3,44. Rezultati so pričakovani, saj je ta glas po najinem mnenju najslabši in najtežje razumljiv.

Razlog za njihovo oceno je, da se glas še vedno ne sliši naravno, čudno izgovarja določenih črk, govori hitreje in je zato bolj razumljiv. Navedli so tudi, da je osnovna hitrost govora dobra in da se v primerjavi s prejšnjim besedilo veliko bolje razume.



Graf 5: Ocene drugega glasu

6.6 Kako razumljiv se vam zdi 3. posnetek?

S šestim vprašanjem sva povprašala po oceni glasu, ki je bil narejen s pomočjo PMM modela in 400 posnetki (verzija 2.0). Povprečna ocena je 3,64. Ta glas je precej naprednejši od glasu v 1. posnetku, saj glas sedaj nakazuje na padajočo in naraščajočo intonacijo na koncu besed/stavkov.

Razlog za njihovo oceno je, da včasih začne brati s preveč robotskim glasom, ampak je veliko bolj tekoči kot prejšnji.



Graf 6: Ocene tretjega glasu

6.7 Kako razumljiv se vam zdi 4. posnetek?

S sedmim vprašanjem sva povprašala po oceni glasu, ki je bil narejen s pomočjo unit selection modela in 400 posnetki (verzija 3.0 in s spremenjenimi nastavitvami pri izdelovanju). Povprečna ocena je 3,68. Rezultati so malo presenetljivi, da je bil ta glas najbolje ocenjen. Je pa res, da je ta glas najbližje človeškemu.

Razlog za takšno oceno je, da so besede bolje poudarjene, hitrost je ravno prava, barva glasu je bolj doživeta, vendar še vedno je premalo premora med besedami.



Graf 7: Ocene četrtega glasu

6.8 Preizkus sintetizatorja

Nazadnje sva dala anketirancem možnost, da sami preizkusijo sintetizator s svojim besedilom. Komentar so napisali trije in povedali, da sta naglaševanje besed in izgovorjava nenaravna, nekateri izmed njih pa so napisali spodbudne komentarje:

»Super, da se s tem ukvarjate. Zelo si želim, da bi še izboljšali govor. To bi bil velik napredek tudi za naš produkt Feelif. Vse dobro, lp Katarina«

»Odlično!«

»Kul! Res super.«

»Menim, da v trenutnem stanju, torej stanju razvoja deluje sintetizator dobro in je njegovo branje razločno, vendar še ni dovolj dobro, za komercialno rabo.«

7 Razprava

Izdelovanje sintetizatorja je vsekakor zapleten projekt, ki ni le časovno zahteven, temveč tudi miselno. Dandanes gre večina zaslug odprtokodnim projektom, s katerimi lahko naredimo marsikaj. Med drugim sta to MaryTTS in Tacotron. Z njimi lahko posameznik z željo in nekaj znanja ter časa naredi dokaj kvaliteten sintetizator govora.

7.1 1. hipoteza

Prvo hipotezo »Z vsaj enim izmed uporabljenih orodij, nama bo uspelo izdelati slovenski sintetizator govora« lahko **potrdiva**, saj nama je uspelo izdelati kar tri glasove s pomočjo MaryTTS orodja.

7.2 2. hipoteza

Drugo hipotezo »Z globokim učenjem (ang. deep learning) bova dobila bolj razumljiv glas kot z drugimi metodami« lahko **ovrževa**, saj nama v času, ki nama je bil na voljo, s tem načinom ni uspelo narediti razumljivega glasu.

7.3 3. hipoteza

Tretjo hipotezo »Končni glas bo razumljiv« lahko **potrdiva**, saj je končni glas precej razumljiv in do neke mere uporaben, morda še ne za daljše poslušanje, a nenazadnje je večina anketirancev glasove vendarle ocenila pozitivno.

8 Zaključek

Sinteza govora je precej kompleksen postopek, še posebej, če bi izdelovali sistem od začetka. Odprtokodni sistemi nam omogočajo enostavno izdelovanje posodobljenih oz. novih izdelkov.

Glasove, ki sva jih ustvarila, bi lahko izboljšala. Glasovi pri MaryTTS naglašujejo nekatere besede napačno, ker so izgovori v leksikonu napisani napačno oz. v drugačnem kontekstu. Da bi ta problem odpravila, bi lahko posodobila izgovorjavo teh besed v slovarju, kar bi vodilo v boljši izgovor besed s pravim naglasom.

Pri globokem učenju na žalost nisva dobila uporabnih rezultatov. Morala bi podrobneje pregledati nastavitve Tacotron-a in TensorFlow-a ali uporabiti kakšen drug algoritem za izdelovanje glasu.

Meniva, da je bila ta raziskovalna naloga uspešna, saj sva se naučila marsikaj o slovenskem jeziku, ukazih na Linux operacijskem sistemu, o programskem jeziku Java, o globokem učenju itd.

9 Zahvala

Za pomoč pri realizaciji raziskovalne naloge se iskreno zahvaljujema mentorici gospe Valentini Hrastnik, strokovnemu mentorju mag. Boštjanu Resinoviču in sošolcu Binetu Lipušu, ki nama je pomagal pri izvajanju procesov, za katere sva potrebovala močnejši računalnik. Prav tako se zahvaljujema viš. pred. dr. Robertu Rozmanu s Fakultete za računalništvo in informatiko, ki nama je svetoval pri uporabi SloLeks-a 2.0., in podjetju Arctur, ker sva lahko na njihovem superračunalniku poganjala globoko učenje.

10 Viri in literature

2019. Deep learning. *Wikipedia*. [Online] 2019. https://en.wikipedia.org/wiki/Deep_learning.

Dobrovoljc, Kaja, Krek, Simon in Erjavec, Tomaž. 2017. The Sloleks Morphological Lexicon and its Future Development. 2017.

Golob, Tadej. 2019. *Jezero*. [nast.] Sebastian Cavazza. Ljubljana : s.n., 2019.

Gros, Jerneja. 2000. *Samdejno tvorjenje govora iz besedil*. Ljubljana : Založba ZRC, (ZRC SAZU), 2000. 961-6358-21-9.

Ito, Keith. 2017. Tacotron. *A TensorFlow implementation of Google's Tacotron speech synthesis with pre-trained model (unofficial)*. 2017.

Microsoft. 2019. C# programming guide. *C# programming guide*. [Online] 20 January 2019. <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>.

Murray, David, [writ.]. 2019. *How Speech Synthesizers Work*. 8-Bit Guy, 2019.

Open source voice creation toolkit for the MARY TTS Platform. **Schröder, Marc, et al. 2011.** 2011. International Speech Communication Association. pp. 3253-3256.

Robida, Nejc. 2013. *Sinteza govora in Govorec 3*. Ljubljana : s.n., 2013.

Sharma, Sagar. 2017. What the Hell is “Tensor” in “TensorFlow”? [Online] September 6, 2017. <https://hackernoon.com/what-the-hell-is-tensor-in-tensorflow-e40dbf0253ee>.

Žganec Gros, Jerneja. 2018. *eBralec – sintetizator govora za slovenščino*. Ljubljana : s.n., 2018.

Žganec Gros, Jerneja, in drugi. 2016. *Sintetizator govora za slovenščino eBralec*. Ljubljana : /, 2016.

Viri slik:

- Slika 1: <https://i2.wp.com/120years.net/wordpress/wp-content/uploads/61aG6NtaLZL.jpeg>
- Slika 2: [https://en.wikipedia.org/wiki/Speak_%26_Spell_\(toy\)#/media/File:Speak_&_Spell_\(original_style\).jpg](https://en.wikipedia.org/wiki/Speak_%26_Spell_(toy)#/media/File:Speak_&_Spell_(original_style).jpg)
- Slika 3: <https://github.com/marytts/marytts/wiki/New-Language-Support>
- Slika 4: Lasten arhiv
- Slika 5: Lasten arhiv
- Slika 6: <https://github.com/keithito/tacotron/issues/144>
- Slika 7: Lasten arhiv

Vsi elektronski viri so bili dostopni še 4. 6. 2020.

11 Priloge

11.1 Anketa

Spoštovani!

Sva Marko Vrečer in Nik Grebovšek, dijaka 4. letnika ŠCC KER, smer računalniški tehnik in delava raziskovalno nalogo na področju sinteze govora.

Prosiva vas, da rešite spodnjo anketo.

(Linki se odprejo na novi strani. Ko končate s poslušanjem, v brskalniku pritisnite tipko nazaj. Odgovori v anketi se bodo shranili).

1. Koliko ste stari?

- 14 ali manj
- 15-16
- 17-18
- 19-20
- 21-22
- 22-23
- 24-25
- Ostalo

2. Ali ste že kdaj slišali za sintezo govora?

- Da
- Ne

3. Poznate katerega izmed slovenskih sintetizatorjev govora? (neobvezno)

- eBralec
- Microsoft Lado
- Proteus

4. Kaj po vašem mnenju na splošno najbolj vpliva na slabše razumevanje govora/pogovora?

- Hitrost govora
- Narečje
- Govorne napake
- Ostalo

1. Posnetek

Posnetek: <https://l.nikigre.si/ibXg>

5.Kako razumljiv se vam zdi ta posnetek?

6.Zakaj?

2. Posnetek

Posnetek: <https://l.nikigre.si/DjU8>

7.Kako razumljiv se vam zdi ta posnetek?

8.Zakaj?

3. Posnetek

Posnetek: <https://l.nikigre.si/jams>

9.Kako razumljiv se vam zdi ta posnetek?

10.Zakaj?

4. Posnetek

Posnetek: <https://l.nikigre.si/WDa9>

11.Kako razumljiv se vam zdi ta posnetek?

12.Zakaj?

Zaključek

13.Če želite, lahko sintetizator preizkusite tukaj: <https://l.nikigre.si/Sintetizator> in opišete svoje mnenje.

11.2 Izjava mentorice

IZJAVA*

Mentorica Valentina Hrastnik v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Sinteza govora, katere avtorja sta Nik Grebovšek in Marko Vrečer:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 5. 6. 2020



Podpis mentorja

Podpis odgovorne osebe

*

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

11.3 Izjava mentorja

IZJAVA*

Mentor Boštjan Resinovič v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Sinteza govora, katere avtorja sta Nik Grebovšek in Marko Vrečer:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 4.6.2020

žig šole

Podpis mentorja

Podpis odgovorne osebe



*

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.