TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Rasmus Lellep

Mitmekeelne kõnesüntees eesti keelega

Bakalaureusetöö (9 EAP)

Juhendaja(d): Liisa Rätsep, MSc

Tartu 2021

Mitmekeelne kõnesüntees eesti keelega

Lühikokkuvõte:

Vajadus kõnesünteesi, sealhulgas mitmekeelse sünteesi järele järjest kasvab. Kui varasemalt põhines süntees manuaalselt seatud keelelistele tunnustele või inimese kõnesüsteemi osade jäljendamisele, siis viimastel aastatel on palju üritatud leida kokkuhoidlikumaid ja vähem eeltööd nõudvaid sünteesija tegemise viise. Need lähenemised on suures osas masinõppe, eelkõige tehisnärvivõrkude põhilised. Töös anti ülevaade osadele sellistele lähenemistele, kus on lisaks keskendutud ka mitmekeelsele sünteesile. Ühelgi mitmekeelsel lahendusel pole aga veel tuge eesti keelele. Käesoleva töö raames koostati esimene eesti keelega töötav mitmekeelne tekstist kõne sünteesija. Kokku töötab valminud sünteesija 11 keelega. Töös on kirjeldatud sünteesija tegemiseks vajaminevaid etappe. Koostatud näidislausete põhjal viidi läbi mudeli eesti keele kvaliteedi hindamine. Lõpuks toodi välja võimalikke parandusi, millega saaks sünteesija genereeritud eestikeelse kõne kvaliteeti veel tõsta.

Võtmesõnad:

Kõnesüntees, mitmekeelsus, tehisnärvivõrk

CERCS: P170 Arvutiteadus, P175 Informaatika

Multilingual speech synthesis with Estonian

Abstract:

The need for speech synthesis, including multilingual synthesis is continuously growing. While earlier approaches of speech synthesis were based on manually composed linguistic features or imitating the components of human speech system, in the recent years there have been a lot of attempts of creating solutions that require less preparation and are less demanding. These solutions are largely based on machine learning, primarily artificial neural networks. This thesis gives an overview to some of the aforementioned solutions, which also focus on multilingual synthesis. None of the multilingual solutions have the support for Estonian. In the practical part of this thesis, the first multilingual text-to-speech synthesizer with support for Estonian was made. The synthesizer works with 11 languages in total. The thesis describes the steps of creating this synthesizer. With composed test sentences, the synthesizer's Estonian speech quality was evaluated. Finally, some adjustments that could further improve the Estonian speech quality were proposed.

Keywords:

Speech synthesis, multilingual, artificial neural network

CERCS: P170 Computer science, P175 Informatics

Sisukord

1. Sissejuhatus

Kõnesüntees on kirjakeele teisendamine sünteetiliseks inimkõneks. Eriti oluline on kõnesüntees ja sellega seotud lahendused nägemispuudega ning kõnevõimetutele inimestele. Kõne sünteesimist kasutatakse olulisel määral näiteks navigeerimis- ja teavitussüsteemides, keeleõppemeetodites, juturobotites ning operatsioonisüsteemides. Ühed tuntuimad tehnoloogialahendused, mille üks tähtsaid komponente on kõnesüntees, on Apple'i Siri ja Google Assistant. Praeguseks on aga peaaegu igal infotehnoloogia valdkonnas tegutseval suurfirmal omad kõnesünteesimudelid ja nende arenduseks oma meetodid. Kõnesünteesiga sünteesitud kõne on veel enamjaolt üpris eristatav ja jääb päris inimkõnele tajutava kvaliteedi poolest alla, kuid see on kokkuhoidlikuks lahenduseks firmadele nii mahu kui ka tööjõu poolest. Seda sellepärast, et lihtsasti arusaadavat kõnet on sünteesimudeli olemasolul võimalik sünteesida väga väikse ressursivajadusega ning tihti ei ole tarvilik, et kõne oleks laitmatu kvaliteediga, vaid piisab sellest, et kõne oleks inimesele arusaadav.

Tehnoloogia arenguga globaliseerub maailm järjest enam, sestap on palju tavapärasemaks saanud keelte vaheldumine tekstides [1]. Sagedasimad kokkupuuted mitmekeelsete tekstidega on navigatsioonisüsteemid [2, 1], kus juhised küll antakse kasutaja valitud keeles, kuid osad sõnad (näiteks kohanimed) võivad olla muus keeles, ja kõnekeelsed tekstid [1], kuhu kombineeritakse sisse järjest rohkem mitme eri keele üldtuntumaid väljendeid. Üha rohkema mitmekeelse teksti esiletulekuga tekib vajadus ka mitmekeelsele kõnesünteesile. Selliseid kõnesünteesimudeleid on küll juba tehtud, kuid ükski neist ei sisalda eesti keelt.

1.1 Eesmärk

Teadustöö raames tehtud praktilise töö eesmärgiks on luua kõige esimene ka eesti keelega töötav kõnesünteesi mudel, mis suudab koodivahetusega tekstist sünteesida võimalikult kvaliteetset ja tekstile vastavat kõne. Koostatud testandmete peal viiakse läbi küsitlus, et hinnata tulemuste kvaliteeti. Olemasolevad andmed ning integratsioon kasutatava koodiga on järgmiste keeltega: mandariini, hollandi, soome, prantsuse, saksa, kreeka, ungari, jaapani, vene ning hispaania. Eelmainitud keeltega on kasutatavas arhitektuuris treenitud mudel juba olemas. Käesolevas töös valmib mudel, mis on treenitud eelmainitud keelte ning lisaks eesti- ja inglisekeelsete andmete põhjal. Tulemuseks on mudel, mis suudab sünteesida kõne, sõltumata sellest, milliseid ja kui palju eelmainitud keeli on lauses kasutatud.

2. Mõisted ja terminid

Koodivahetus – eri keelte vaheldumine tekstis.

Mean opinion score (MOS) – isikute subjektiivsete kindlal skaalal antud süsteemi kvaliteedi hinnangute aritmeetiline keskmine.

Kvantimine – protsess, mille käigus jagatakse signaali mõõtepiirkond lõplikuks arvuks vahemikeks ning seejärel ümardatakse analoogväärtused jagatud vahemike järgi digitaaalväärtusteks.

Mel skaala – helikõrguste skaala, mille võrdsed vahemikud on inimkõrvale tajutavalt lineaarsed.

Spektrogramm – helisignaali ajamuutlikkuse visuaalne kujutis.

Vokooder – koodisignaali kõnesignaaliks või vastupidi muundav seade.

Sämplimine - protsess, mille käigus muudetakse pidev analoogsignaal diskreetsetest näitudest koosnevaks digitaalsignaaliks.

Sämplimissagedus – digitaalse signaali (mitte analoogne/pidev, vaid järjestikustest üksiknäitudest koosnev) näitude arv sekundis.

Foneem – häälikusüsteemi väikseim osa, mis eristab ühe sõna tähendust teisest. Foneemid erinevad keelepõhiselt.

Difoon – kõne lõik, mis sisaldab esimese foneemi püsivat osa, esimese foneemi üleminekut teisele foneemile ning teise foneemi püsivat osa.

Epok – üks terve treeningtsükkel üle kõigi andmete.

Must kast – süsteem, millel on teada vaid sisendid ja väljundid, sisemisi töötlusi ei ole võimalik vaadelda.

Hüperparameeter – parameeter, mis seatakse mudelile enne selle treenimise algust.

3. Taust

Käesolevas peatükis on lühidalt kirjeldatud tehisnärvivõrke, kõnesünteesi olemust, selle erinevaid meetodeid ning lähenemist mitmekeelsele kõnesünteesile. Lõpuks tuleb juttu ka kõnesünteesi lahenduste hindamisest.

3.1 Tehisnärvivõrgud

Tehisnärvivõrgud on arvutuslikud süsteemid, mis kujundavad ette antud sisendite põhjal enda sisukomponentidele mustreid, mis mõjutavad omakorda ülejäänud komponente. Saadud mustrite põhjal on närvivõrgud võimelised ennustama sisendile vastava lahenduse või ennustuse. Närvivõrkude eelis muude masinõppemeetodite ees on see, et ette ei pea kirjutama ülesandepõhiseid reegleid ega valemeid, vaid õppimine toimub vaid näidete põhjal. Puudusteks on aga suurema andmekogu ning võimsa arvutusliku jõudluse vajadus. Nende probleemide olulisus on aga tunduvalt vähenenud, kuna andmesalvestusruumi mahud, andmekogud on kasvanud ning arvutid on muutunud tunduvalt võimsamaks [3]. Veel üks probleem, mis närvivõrkudel esineb, tuleneb nende keerulisest struktuurist ning musta kasti olemusest. Kui juba õppinud võrk ei anna väljundiks soovitud tulemust, siis on põhjuse leidmine väga keeruline.

Visuaalsete sisendite (nagu näiteks pildid või videod) töötlemisel on väga tõhusad konvolutsioonilised närvivõrgud, kuna nende arhitektuur sarnaneb paljuski sellele, kuidas töötab inimese visuaalse tajumise töötlemine ajus [4]. Rekurrentsed närvivõrgud on seevastu väga võimekad järjendite töötlemisel [3]. Need võivad sisendiks võtta suvalise varieeruvusega sisendeid, andes tulemuseks samuti varieeruva pikkusega väljundeid [3]. Rekurrentsed võrgud kasutavad oma varjatud kihtide sisemist mälu nähtud andmete meelde jätmiseks, et hilisemaid sisendeid selle järgi korrektsemalt töödelda osata [3]. Sõlmede väljundid salvestatakse ning õpitakse tagasilevi põhjal, mis tähendab, et väljundid söödetakse mudelile uuesti ette kaalude kohandamiseks [3].

3.2 Kõnesüntees

Kõnesüntees on kirjakeele teisendamine sünteetiliseks inimkõneks. Vastupidiselt traditsioonilisele masinõppele, on tekstist kõne sünteesimisel sisend väljundist märksa väiksema mahuga [5]. Lisaks on erinevaid sobivaid väljundivõimalusi igale sisendile on potentsiaalselt mitu. Näiteks võib sama lauset korrektselt sünteesida mitme erineva hääletooniga,

helitugevusega, aktsendiga, kiirusega. Need omadused teevad kõne sünteesimise üsna keeruliseks ülesandeks. Siiski on nüüdseks kasutusel erinevaid meetodeid, igal esinevad oma eelised ja puudused.

3.2.1 Kõnesünteesi meetodid

Üks lähenemine tehisliku hääle saamisele on artikulatoorne süntees. See on meetod, mis sünteesib heli inimese kõneelundeid ja nende tegevust võimalikult täpselt imiteerides. Antud meetod on potentsiaali poolest parim: piisavalt täpse modelleerimisega on võimalik teha kõnemudel, mis jäljendab inimkõne ideaalselt [6]. Sellise sünteesija mudeli koostamine on aga väga keeruline, kuna näiteks juba keele liikumist on peaaegu võimatu ideaalselt modelleerida [7]. Lisaks nõuab artikulatoorne süntees tohutut arvutuslikku jõudlust. Modelleerimise keerukuse ja tohutu jõudlusevajaduse tõttu ei ole artikulatoorne süntees eelistatav, mistõttu otsustatakse kasutada palju vähem eeltööd nõudvaid süsteeme. [8]

Veel üks viis, kuidas sünteesida tehislikku kõne on formantsünteesi abil. Formantsüntees toimib allika-filtri mudeli põhjal: imiteeritakse häälekurdude võnkumist, et tekitada heli, ja kõnetrakti resonantssagedusi, et filtreerida helisagedused vastavaks. Nii toimiv mudel ei jäljenda kõneelundeid täielikult, samaks jääb vaid kõnesüsteemi üldine struktuur. Formantsüntees oli 20. sajandi lõpukümnenditel kõige laiemalt kasutatud sünteesimeetod. [8]

Kõige lihtsam on koostada loomulikku ja arusaadavat häält kompilatiivse sünteesi abil. Kompilatiivne süntees on tihti kasutusel piiratud kõnesünteesisüsteemides, mis tähendab, et sünteesida on vaja vaid konkreetseid sõnu ja nende kombinatsioone. Mida kitsama ulatusega sünteesi vaja on, seda konkreetsemaid andmeid saab kasutada, näiteks tavalise foneemidest või difoonidest koosneva andmestiku asemel kasutada silpide, täissõnade või kogunisti täislausete andmestikku, ning seeläbi saab väljundkõne loomulikumaks ja lihtsamini reguleeritavaks [7]. Laialdasemalt kasutamiseks ei ole aga sedasi tehtav süntees otstarbekas, kuna mälu vajadus on võrreldes alternatiivsete meetoditega palju suurem. See probleem süveneb, kui tahta koostada mitmekeelset või mitmehäälset mudelit, kuna iga häälele ja keelele on vaja omaette täismahus andmeid. [8]

Head kõne suudab sünteesida ka statistiline parameetriline süntees, mille tuntuim meetod on süntees Markovi peitmudelite abil. Markovi peitmudeli süntees toimub kahes etapis: treenimine ja sünteesimine. Treenimisel eraldatakse kõneosadest parameetrid andmebaasi Markovi peitmudelitena. Sünteesimisel otsitakse andmebaasist sõnadele vastavad

peitmudelid, mille parameetreid kasutatakse kõne sünteesimiseks. Sellise meetodi miinuseks on aga üldiselt halb kõne loomulikkus. [7]

Tänapäeval enim kasutatav meetod piiramata kõnesünteesiks on närvivõrgud [9]. Hästi koostatud närvivõrgud nõuavad võrdlemisi väiksemat andmestikku eelnevatest lahendustest, et saavutada ligilähedast kvaliteeti [10]. Lisaks suudavad osad närvivõrguarhitektuurid õppida hääli kloonima [11, 12, 13]. See tähendab, et süsteem oskab eri häältes olevaid häälduste kombinatsioone kasutada iga häälega, mis andmetes esinenud on.

Eelmainitud sünteesimeetodite (välja arvatud närvivõrkude) implementatsioonid on aga väga suures enamuses tehtud vaid ühekeelset sisendit eeldades. Mitmekeelse sisendi puhul tuvastatakse või lastakse valida üks keel ning üritatakse terve tekst selle keele häälduspärade järgi kõneks sünteesida. Koodivahetusega kõne puhul tekivad sellise meetodiga hääldus- ja rõhuvead, teistsuguse tähestiku puhul võib osa üldse sünteesimata jääda.

3.2.2 Mitmekeelne kõnesüntees

Üks võimalus mitmekeelse teksti kõneks teisendamiseks ühekeelsete sünteesijate abil on teksti ühekeelseteks osadeks jagamine ning nende eraldi sisendiks andmine ühekeelsetele mudelitele, lõpuks väljundite kokku kleepimine. Sellega esineb aga palju probleeme. Esiteks on selliselt sünteesimine ressursi- ja ajamahukas, kuna tehakse mitu sünteesi ühe asemel. Teiseks ei ole tulemusena saadud kokku kleebitud tekst loomuliku ilmekuse ja tooniga, samuti puudub sõnadel kontekstipõhine rõhk ja toon. Lõpuks, kuna eri keelte treeningandmetel on peaaegu alati erinevad kõnelejad, siis sünteesitud kõne hääl vaheldub koos keelega, mis teeb väljundi veel ebaloomulikumaks.

Hiljutised mitmekeelse kõne sünteesijad [1, 11, 12] lahendavad osad eelmainitud probleemid. Need on kiiremad ning treenimiseks ei ole vaja mitmekeelseid andmeid, vaid piisab mitme keele ühekeelsetest andmetest. Iga hääl on vajadusel sünteesiks kasutatav ka teistes esinevates keeltes. Tulemusena suudetakse sünteesida loomulikulähedast häält koodivahetusega tekstist. Parimad neist on isegi võimelised subjektiivse kvaliteedi poolest konkureerima päris inimese kõnega [12].

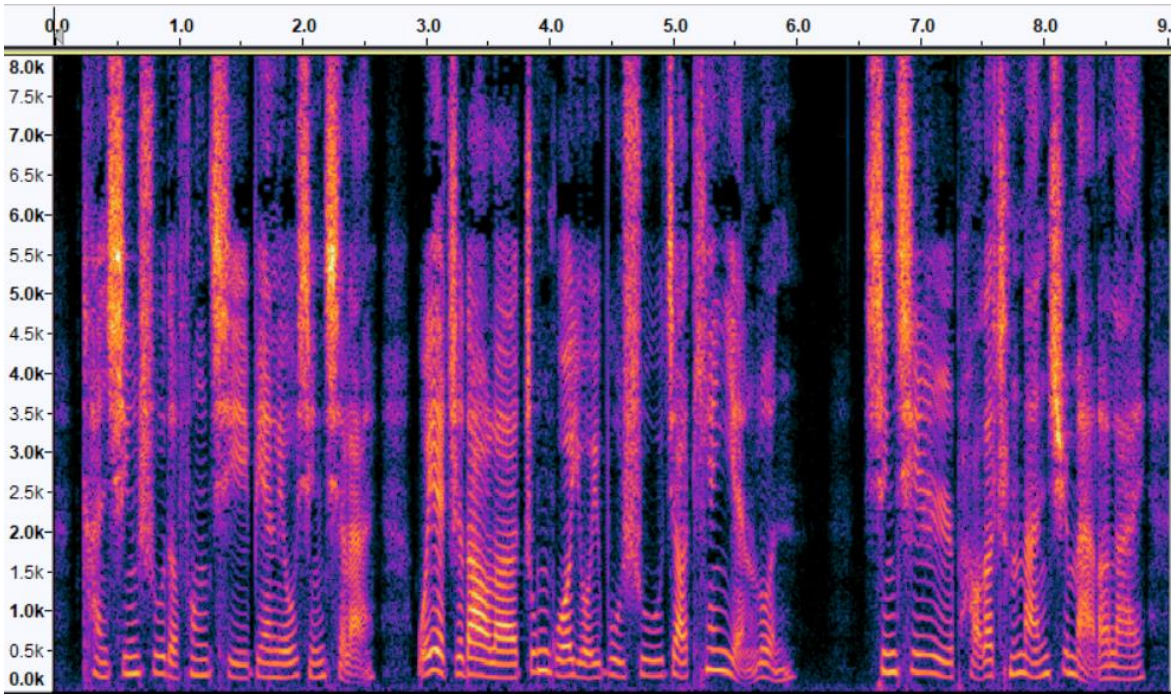
Mitmekeelsete sünteesija tegemisel on väga oluline arvestada sinna planeeritud keeltega. Keeled, millel on maailmas kõige rohkem kõnelejaid, on tihti andmetel treenitavates kõnemudelites ka esindatud, kuna erineva mahu ja formaadiga andmekogud nendes keeltes

on hõlpsasti leitavad. Kui aga tegemist ei ole nii populaarse keelega, võib olla andmete kogumine raske ning isegi kui andmed on olemas, siis ei pruugi olla valikuvõimalust andmekogude vahel. See raskendab märkimisväärselt ebapopulaarsete keeltega mudeli koostamist.

Tihti esineb kõnesünteesi väljundiga probleeme. Mõnikord jääb väljundiklipist mõne sõna või tähe hääldus puudu, sõnade vahel võivad esineda ebaharilikult pikad pausid [11]. Kuna mitmekeelses sünteesis valitakse lause jaoks ära üks treeningandmetes sisaldunud häältest, mis on ühekeelsetes andmetes esindanud vaid ühte keelt, siis võib juhtuda, et osa lausest, mis on kõneleja keelest erinevas keeles, on hääldatud kummalise aktsendiga või sootuks valesti. Vahel võib ka teisest keelest lauseosa tekitada raskusi nii, et väljundisse jääb üle toodud hääle tunnuseid.

3.2.3 Spektrogramm ja vokooder

Spektrogramm on graafiline kujutis, mis kirjeldab helisageduste esinemisi ajas ning nende sageduste tugevust. Paljud masinõppe põhilised kõnesünteesisüsteemid kasutavad vaheetapina spektrogramme, kuna neid on masinõppmudelil lihtsam ennustada kui valmis helifaili, neid on helist otse väga lihtne genereerida ning infokadu heli kohta on väga väike. Sellistes süsteemides on esimene pool sünteesist tekstist spektrogrammi genereerimine – erinevate helisageduste ning nende tugevuste ennustamine vastavatel ajahetkedel.



Joonis 1. Visuaalne kujutis helifaili spektrogrammist. Vasak-parem telg iseloomustab aega sekundites, üles-alla telg sagedusi hertsides ning värvi erksus selle sageduse tugevust.

Teine pool on spektrogrammist helisignaali saamine vokooderi abil ning helifaili genereerimine. Vokooder võib olla universaalne, spetsiaalselt treenitud või algoritmiline (näiteks Griffin-Lim algoritm [14]). Kuna helist spektrogrammi genereerimine on kiire matemaatiline protsess, on vokooderi treenimiseks lähteandmeid saada lihtne.

3.2.4 Hindamine

Tekstist kõneks sünteesija üks väljakutsetest on selle kvaliteedi hindamine. Märkimisväärse osa selliste implementatsioonide kvaliteedi hindamistest toimub *mean opinion score* meetodil. See meetod on subjektiivne, kuna sõltub inimeste hinnangutest hääle kvaliteedile. Tihti on isegi kahe võrreldava hääle hindajad olnud erinevad, mis teeb meetodil saadud tulemuste võrdlemise veelgi ebamäärasemaks. Seega, kui mitu erineva kõnesünteesimudeli sünteesitud väljundit on andnud üksteisele lähedasi *mean opinion score* tulemusi, ei saa selle põhjal kindlalt väita, milline neist tegelikult parim on.

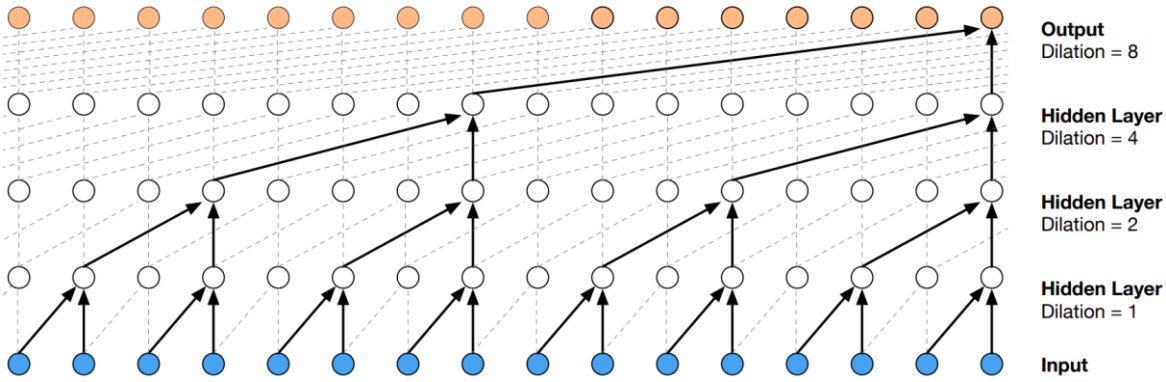
Mean opinion score meetodi teostamiseks peab leidma iga kvaliteedihinnagu saava keele jaoks inimesed, kes seda keelt oskavad, eelistatavalt emakeelena. Mitmekeelse mudeli jaoks oleks hindajateks ideaalsed inimesed, kes oskavad testitavatest keeltest mitut. Mida rohkem inimesi mudelit hindab, seda täpsemaks hinnang läheb. Et hindamisel võimalikult palju mudeli võimekust või võimetust arvesse võetaks, koostatakse hindamiseks palju erinevaid sünteeskõne klippe. Kuna mitmekeelseid tekste esineb vähe, siis tuleb mitmekeelse mudeli hindamiseks testlaused koostada. Kõiki neid etappe arvestades on *mean opinion score* kallis ja aeganõudev hindamismeetod.

4. Seotud tööd

Peatükis on tutvustatud erinevaid närvivõrkudel põhinevaid olemasolevaid lahendusi, mis on käesoleva töö praktilise osa koodibaasile eelkäijateks või oluliseks toeks.

4.1 WaveNet

Üks murrangulisi närvivõrguarhitektuure helisünteesis on 2016. aastal avaldatud konvolutsiooniline närvivõrk WaveNet [15]. WaveNet teeb ennustusi üksiku signaalinäidu kaupa, kus iga ennustatav diskreetne signaalinäit on tingitud kõigist varasematest ennustustest. Arhitektuuri konvolutsioonid on põhjuslikud (*causal*) ehk ainult edasi levivad (mitte ühegi ajahetke ennustus ei saa sõltuda järgnevate ajahetkede ennustustest) ning eksponentsiaalse laiendamisega (*dilation*), kasvades iga kihiga kahekordselt 1-st kuni limiidini ning siis seda korrates, et teha iga ennustus sõltuvaks võimalikult pikast eelnevate ennustuste vahemikust. WaveNet saavutas märgatavalt paremaid tulemusi, kui ükski selleaegne kompilatiivne või parameetrite tingimustel põhinev sünteesija. [15]



Joonis 2. Kujutus põhjuslikest eksponentsiaalse laiendamisega konvolutsioonidest [15].

Kuigi WaveNeti abli sünteesitud heli on väga hea kvaliteediga, on hilisemates lahendustes toodud välja ka selle nõrku kohti ning üritatud neid kõrvaldada. Esiteks, WaveNet on aeglane juba oma loomu poolest, kuna teeb ennustusi kõigi eelnevate ennustuste põhjal ning ennustab vaid ühe signaalinäidu kaupa [5]. Teiseks, kuna kõnesünteesiks on mudelile vaja sisendiks anda andmed, mida on keeleliste tunnuste põhjal juba kohandatud, siis ei ole WaveNet otsast-lõpuni (*end-to-end*) süsteem ja seda ei saa eraldiseisvalt tekstist kõneks sünteesiks kasutada [5].

4.2 WaveRNN

WaveRNN vokooder on kompaktne rekurrentne närvivõrk, mis suudab vastanduda väga hea kvaliteediga WaveNetile. Kuna süsteem on tunduvalt kompaktsem, suudab

WaveRNN genereerida heli WaveNetist tunduvalt kiiremini. Lisaks toodi välja uus genereerimise skeem. Selle järgi jaotatakse spektrogrammijärjend osadeks, saades nii genereerida mitu signaalinäitu paralleelselt, kiirendades protsessi veelgi. [16]

4.3 Tacotron

Ühtsel otsast-lõpuni (*end-to-end*) süsteemil on eeliseid mitmest komponendist koosneva süsteemi ees. Esiteks on mudeli treenimine lihtsam, kuna andmed tuleb sisendiks anda vaid ühe korra. Lisaks on ühtsel süsteemil ka rangem struktuur, millega saab vältida olukorda, kus mitme eri komponendi vead kuhjuvad. [5]

2017. aastal avaldasid Wang jt [5] uue *end-to-end* närvivõrgumudeli kõnesünteesiks - Tacotroni. Kui WaveNeti osa kõnesünteesis on vaid vokooder ja akustiline mudel, siis Tacotron asendab kõik etapid tähemärkidest kuni lineaarskaalas spektrogrammikaadriteni, mis teisendatakse Griffin-Lim algoritmi [14] kaudu edasi lainekujuks (WAV-helifailiks). Tacotron on oma kaadripõhise ennustuse tõttu ka tunduvalt kiirem kui WaveNet. Tacotron töötab täiendatud rekurrentse närvivõrgu *sequence-to-sequence* [17] baasil, millele on lisatud ka tähelepanu mehhanism. Standardse *sequence-to-sequence*'i kooderi-dekooderi asemel, kus terve lause kodeeritakse peidetud kihti ühe fikseeritud vektorina, kodeeritakse lause hoopis vektorite järjendiks, millest valitakse dekodeerimise lõpus sobivaim alamhulk [18]. Sellise lähenemisega kaob vajadus mahutada kodeeritud vaheolekut ühte fikseeritud pikkusega järjendisse, süsteem muutub paindlikumaks ja kohanemisvõimelisemaks [18]. [5]

4.4 Tacotron 2

Närvivõrkude põhistest lahendustest saavutab *mean opinion score*'iga mõõdetult kõrgeid tulemusi J. Shen jt [19] poolt loodud arhitektuur Tacotron 2. Kuigi süsteem põhineb samamoodi *sequence-to-sequence* [17] arhitektuuril, siis võrreldes esimese Tacotroniga on seda kohandatud nii, et sisendiks antud tähevektoritele oleks mudelis vastavaks väljundiks mel skaalas spektrogrammid. Mel skaalas spektrogrammid on mahu poolest väiksemad, seega on neid vähenõudlikum töödelda, ning hea vokooderi korral jääb neid kasutades sünteesitud heli samale kvaliteeditasemele. Kuna Griffin-Lim algoritmiga koostatud kõnelõigud kõlavad robotlikult ning sisaldavad tajutavaid artefakte, siis võeti kasutusele ka loomulikumat kõne väljastav vokooder, mida treeniti eraldi. Nagu esimese Tacotroni [5] avaldamisel ka ära mainitud on, siis oli Griffin-Lim algoritmi kasutamine vaid kohatäide selleni,

kuni see asendatakse parema lahendusega. Uus lahendus saadi WaveNeti vokooderi struktuuri lihtsustamise teel, kahandades võrgu kihte 30-st vaid 12 peale, vähendades laiendamist, mistõttu vähenes ennustust mõjutav eelmiste ennustuste vahemik 256 millisekundi pealt vaid 10.5 millisekundini, mis kiirendas kordades sünteesimise kiirust. [19]

4.5 Mitmekeelsed lahendused

Kõige paremaid kvaliteethinnanguid saanud mitmekeelse kõne sünteesija on Tacotron 2 edasiarendus Y. Zhang jt (Google) poolt [12], mille mõningad väljundid sõltuvalt keelest, on hinnatud isegi võrdväärseks päris kõnega. Tacotron 2 arhitektuuri on täiendatud lisasisendesitustega keele ja hääle jaoks, eraldi kodeerijaga prosoodia ja müra tuvastamiseks ning kontrollimiseks ja lõpuks ka hääle klassifitseerijaga, et väheste häältega keeltel hääle ja keele tunnused ei ühilduks ja seega neid eraldi töödeldaks. Vokooder on samamoodi treenitud eraldi, kuid WaveNeti asemel on kasutatud WaveRNN-i. [12]

Y. Zhang jt [12] töösse on integreeritud inglise, hispaania ja mandariini keel. Kuna need on teatavasti kolm kõige rohkem räägitud keelt maailmas ning Google'i omanduses on üks maailma suurimaid andmekogusid, siis olid autorid väga võimekad kogumaks kokku kvaliteetseid andmeid. Vaatamata parimatele kvaliteedihinnangutele, ei ole käesolevas töös selle lahenduse lähtekoodi siiski kasutatud, kuna antud teostuse kood ei ole avalik. Lisaks, kuna selle arhitektuuri treenimiseks kasutatud andmed samuti avalikud ei ole, siis ei pruugi parimad *mean opinion score* näitajad tuleneda vaid arhitektuurist, vaid ka kasutatud andmete kõrgest kvaliteedist.

T. Nekvinda jt [11] on loonud mitmekeelse kõnesünteesi implementatsiooni, mille baasiks on samuti Tacotron 2. Süsteem on lihtsasti skaleeruv uutele keeltele ja häältele. Tacotron 2 on võetud aluseks. Tacotroni rekurrentse kodeerija asenduseks sai mitu keelepõhist konvolutsioonilist teksti kodeerijat [20]. Lisati juurde kontekstipõhine parameetrite generaator E. A. Platanios jt [21] eeskujul, mis genereerib vastavalt keele omadustele kodeerijale parameetrid. Iga kodeeritud osa, mis dekodeeritakse spektrogrammideks, põimitakse enne dekodeerimist ka hääle identiteediga, mis võimaldab mitmekeelsust lauses ja keeltevahelist häälte kloonimist. Lisaks sarnaselt Y. Zhang jt [12] lahendusele, tuuakse ka siia süsteemi hääle klassifitseerija, et kodeerijal ei oleks häälepõhist infot ning hääle lisamine toimuks eraldi. [11]

Kuigi viimase lahenduse saavutatud *mean opinion score* ei ole nii kõrge kui eelneva arhitektuuri oma, ei ole tulemused piisavalt suurte erinevustega, et saaks kindlalt väita, et

käesoleva arhitektuuri mudelid samade andmetega treenimisel madalama kvaliteediga kõne sünteesiksid. Võrdlust raskendavad asjaolud on ka, et käesolevas töös on koodi integreeritud rohkemate keelte andmed ja eelnevas lahenduses olulist rolli mänginud inglise keel siit puudub. Keelte rohkus teeb mudeli mitmekesisemaks ja potentsiaalselt targemaks, kuid kuna mõne keele andmeid on tunduvalt vähem kui eelneva mudeli keeltel, siis on palju ebatõenäolisem, et mudel iga keele puhul samale kvaliteeditasemele jõuab. T. Nekvinda jt [11] üritasid implementeerida ka Y. Zhang jt [12] arhitektuuri ning sellega enda mudelit võrrelda, kasutades originaaltööga võrreldes rohkem keeli ja vähem andmeid. T. Nekvinda jt [11] sünteesija saavutas paremaid tulemusi nii tähemärgivigade, sõnade vahele jätmise kui ka subjektiivse soravuse ja täpsuse poolest. Vokooderina on kasutatud WaveRNN-i baasi, mida täiendati vastavaks töötamaks mitmekeelse mudeliga. Käesoleva töö praktilises osas on võetud aluskoodiks just T. Nekvinda jt [11] koostatud arhitektuur.

5. Töö käik

Peatükis kirjeldatakse töö praktilist osa. Praktilise töö peamised osad olid andmete ning arhitektuuriga tutvumine, andmete töötlus sünteesija mudelile, mudeli korrektselt treenima saamine, andmete töötlus vokooderi treenimiseks, vokooderi mudeli treenima saamine ning ettevalmistamine hindamiseks.

5.1 Andmed

Kvaliteetse mudeli treenimiseks ning testimiseks on vaja palju andmeid. Mida kvaliteetsemad on andmed, seda väiksem on andmekogu vajalik suurus saamaks kõrge kvaliteediga kõne sünteesivat mudelit. Lisaks mõjutab mudeli tarkust ka andmete valikuline varieeruvus: andmekoguga, kus esineb vähe eri hääli, kuid kus igal häälel on palju varieeruvaid häälduste kombinatsioone, saab tunduvalt targema mudeli kui sama suure andmekoguga, kus esineb palju erinevaid hääli.

Üheksa keele andmed üheteistkümnest olid CSS10 andmekogust [22], mis koosnes audioraamatute heliklippidest ning neile vastavatest tekstilõikudest. Kõiki neid andmeid oli juba T. Nekvinda jt [11] projektis varasema mudeli treenimiseks kasutatud ja seega oli aluskood koostatud nii, et andmete ja metafailide lugemiseks ei oleks vaja andmetöötlust teha. Et koodi võimalikult vähe muuta ja sellega erinevaid tõrkeid vältida, viidi lisatavad eesti ja inglise keele andmekogud samale kujule ülejäänud kogudega.

Eestikeelseteks andmeteks oli algselt plaanitud järgmised kogumid: Mozilla Common Voice Corpus 6.1 eestikeelsete klippide kogu[1] umbes 14800 faili ja 500 eri häälega, Eesti Keele Instituudi ilukirjandussalvestuste ning üksiklausete salvestuste kogud [23] 22200 heliklipi ja 4 häälega, Eesti Rahvusringhäälingu uudiste salvestuste kogu [24] 11400 klipi ja 7 häälega ning Tartu Ülikooli uudiste salvestuste kogu [25] 41500 klipi ja 4 häälega. *Common Voice* oli avalikult saadaval Mozilla Common Voice'i veebilehel. Inglisekeelseteks andmeteks on valitud avalik LJ Speech andmekogu [26], mis sisaldab 13100 helifaili 50 eri häälega.

[1] Common Voice: http://voice.mozilla.org/

Keel	Heliklippide arv	Kestus kokku
hollandi	6494	14:06:40
hispaania	11111	23:49:49
jaapani	6841	14:55:36
mandariini	2971	06:27:04
prantsuse	8649	19:09:03
saksa	7427	16:42:45
soome	4842	10:32:03
ungari	4514	10:00:25
vene	9599	21:22:10
inglise*	13100	23:55:17
eesti*	63964	113:25:37

Tabel 1. Kasutatud andmekogude pikkused. *andmekogu pole veel arhitektuuris kasutatud.

5.2 Sünteesija mudel

Käesolevas töös on mudeli koostamiseks kasutatud T. Nekvinda jt [11] arhitektuuri[2]. Mudeliks on Tacotron 2-l põhinev närvivõrk. Sisendiks võtab mudel tekstilise lause, mille järel on kirjas eraldajaga eraldatud sünteesitava hääle nimetus ning lõpuks keel või komaga eraldatud keeled koos keeleteksti pikkusega. Nii keel kui ka hääle nimetus peavad olema mudeli treeningandmetes esindatud. Mudeli väljundiks on helifaili spektrogramm.

"Cette requête s'explique par les relations peu conventionnelles que Schrödinger entretient avec les femmes.|french-001|french-68,german-11,french"

Joonis 3. Näidissisend mudelile

5.2.1 Andmete töötlus

Kuna andmeid on palju, siis et treenimine võtaks aega võimalikult vähe ja et see toimuks ilma tõrgeteta, on vajalik kõik andmed viia samale ja võimalikult lihtsalt loetavale kujule. Veel vajalikumaks muutub see siis, kui treenitakse mitu korda, sest isegi kui andmed

[2] Sünteesija arhitektuur: https://github.com/Tomiinek/Multilingual_Text_to_Speech

on viidud kõige optimaalsemale kujule, võib ühel treenimisel kuluda aega mitmest tunnist kuni mitme ööpäevani.

Esmased andmed olid Common Voice'i eestikeelsed heliklipid. Kuna klippide formaat kaasaarvatud sämplimissagedusega ning metafailide struktuur olid erinevad olemasolevate andmekogude omadest ja eri hääli oli palju, siis võttis selle andmekogu töötlemine üpris kaua aega. Klipid olid Common Voice'i poolt juba kategoriseeritud neile antud hinnangute põhjal kolmeks: valideeritud heaks (rohkem häid hinnanguid saanud), valideeritud halvaks (rohkem halbu hinnanguid saanud) ning valideerimata (alla kahe hea või halva hinnanguga). Kasutusse jäid neist vaid heaks valideeritud klipid. Seejärel rühmitati klipid häälte järgi, T. Nekvinda jt [11] eeskujul eemaldati kogust liiga vähe esindatud hääled (alla 50 heliklipi) ning iga järele jäänud hääle klipid toimetati oma kausta. Järgmiseks hõrendati failid MP3 pealt WAV-iks, millega koos viidi ka sämplimissagedus üle mudelile vajaliku 22.05 kHz peale. Lõpuks tehti skripti abil ülejäänud kogudega identne metafail.

Teine eestikeelne andmekogu on tunduvalt suurem kui Common Voice'i andmete kogu, sisaldades samas väiksema arvu hääli. Ka siin tuli sämplimissagedus teisendada ühtseks ülejäänud andmetega. Kuna osad klipid olid selleks teisenduseks liiga lühikesed, tuli need andmekogust kõrvale jätta.

Inglisekeelsete andmetega oli võrdlemisi vähe tööd, vaja oli vaid jaotada helifailid hääle kaupa kaustadesse. Kuna kõik heliklipid olid õiges pikkusevahemikus ning iga hääle kohta oli piisavalt lindistusi, siis ei jäetud sellest kogust andmeid välja.

5.2.2 Treenimine

Et mudel õpiks nii nagu ette nähtud, on vaja treenimise eel seada hüperparameetrid vastavaks. Kuna mudel on sarnane arhitektuuris ühe ettevalmistatud mudeli kategooriaga, siis sai mudeli tegemisel enamiku hüperparameetrite väärtustest muutmata üle tuua. Muudetud said vaid keeli ja andmeid puudutavad muutujad.

Peale igat viite epokit teeb kood kontrollpunkti ehk salvestab selle hetkeni õpitud kaaludega mudeli. Arusaadava, kuid mõningate häädusvigadega kõne sünteesis mudel juba kolmandaks kontrollpunktiks. Kuna treenimine toimus katse-eksitus meetodil, siis aja säästmise eesmärgil olidki esmased treenimised vaid esimeste kontrollpunktideni. Kontrollpunktide tulemusi hinnati, sünteesides mitmes keeles mõne lause ning siis andes need sisendiks

eeltreenitud vokooderi mudelile. Saadud klippe hinnates sai umbkaudselt teada, kui õigesti ja kiiresti mudel hääli sünteesima õpib.

Esmaste treenimiste tulemusi hinnates sai järk-järgult teha ka kohandusi andmetes. Näiteks filtreeriti treeningandmetest välja kõik heliklipid, mille pikkus oli üle 13 sekundi, kuna need põhjustasid mudelil ebamäärasust. Samuti selgus, et Common Voice andmed ei täiendanud teisi eestikeelseid hääli, vaid mõjusid neile rohkem mürana, seega jäeti lõpliku mudeli treenimisel terve Common Voice andmestik kõrvale. Lõplik mudel treeniti TÜ HPC arvutuskeskuses [27] Nvidia Tesla V100 graafikakiirendil. Treenimine kestis 8 päeva, mille jooksul läbiti 85 epokit, selleks etapiks olid mudeli kaalud juba koondunud ja arvutatud veamäärad olid mitmel eelneval kontrollpunktil samad või väga lähedased.

5.3 Vokooder

Käesolevas töös on vokooderina kasutatud WaveRNN implementatsiooni[3] arhitektuuriga treenitud mudelit, mida on täiendatud ning seejärel kasutatud T. Nekvinda jt [11] lahenduses sünteesijale vokooderina[4], saamaks lõppsaaduseks lainekuju helifail. Vokooderi treenimisel ja sellele eelneval andmetöötlusel on võetud eeskuju täiendatud vokooderi projektist ja selles olevatest juhistest.

5.3.1 Andmete töötlus

Esmalt oli vaja treeningandmetest genereerida spektrogrammid, mis saavad vokooderile sisendandmeteks. Selleks oli vaja kõik treeningandmete sisendid läbi töödelda sünteesija mudeliga, et saada sünteesija mudeliga vastavuses (*ground truth-aligned*) spektrogrammid. Spektrogramme saaks genereerida ka otse heliandmetest ilma mudeli abita (koostada *ground truth* spektrogrammid), kuid selliste andmete põhjal treenitud vokooder ei võtaks arvesse mudeli konteksti ja ebatäpsusi. Sünteesija ennustatud spektrogrammide puhul oleksid eelmainitud vokooderi väljundid õiged vaid sellise mudeli korral, mis ennustab ideaalseid, täpselt helile vastavaid spektrogramme. Kuna aga reaalselt ei saa mudeli ennustused ideaalsed olla, siis lähtutakse vokooderi treenimisel spektrogrammidest, mille sünteesija mudel ennustanud on.

[3] Esialgne WaveRNN implementatsioon: https://github.com/fatchord/WaveRNN

[4] Täiendatud WaveRNN implementatsioon: https://github.com/Tomiinek/WaveRNN

Vastavad spektrogrammid koostatud, oli vaja koostada skript liigutamaks kõik äsja genereeritud failid ning algsed andmefailid keele kaupa WaveRNN projekti alamkausta. Failid õigesti struktureeritud, sai koostatud spektrogrammid olemasoleva skriptiga ära kvantida ning lõpuks mel skaalasse töödelda. Kuna käesolev WaveRNN implementatsioon võtab sisendiks just mel skaala spektrogramme, siis rohkemat eeltöötlust vaja ei olnud.

5.3.2 Vokooderi treenimine

Vokooderil kasutati projektis etteantud hüperparameetreid. Treenimiseks polnud seega vaja ühtegi hüperparameetrit muuta. Kuna aga *ground truth-aligned* spektrogrammide genereerimise skript esialgu ei töötanud, oli seda vaja muuta. Selgus, et skript oli mõeldud väga spetsiifilistele andmetele ja genereeritud faile ei sobitatud kokku õigete failinimedega. Probleem ilmnes alles treenimise käigus, mistõttu oli probleemi põhjust leida üsna keeruline. Viga sai lahendatud, kustutades andmete koondmetafailist nii palju ridu, et allesjäänud ridade arv jaguks skripti partii suurusega (*batch size*), milleks oli mudeli keelte arv (11), peale mida tuli manuaalsete käskude läbi sobitada failid õigete nimedega. Vokooder, mida treeniti TÜ HPC arvutuskeskuses [27] Nvidia Tesla V100 graafikakiirendil, salvestas kontrollpunkti peale igat 25 000 sammu. Treenimine kestis neli päeva ja kaheksa tundi, selle ajaga jõuti miljoni sammuni. Miljoni sammuga oli veamäär jõudnud ühtlustuda.

6. Hindamistulemused ja analüüs

6.1 Ettevalmistus

Mudelid koostatud ja treenitud, koostati laused hindamaks, millise kvaliteediga kõne süsteem sünteesib. Kokku oli mudelis eestikeelsete andmete põhiseid hääli 14, kuid osad neist andsid valideerimisel väga ebaloomuliku väljundi, seega neid hääli hindamisse ei kaasatud. Ülejäänud seitsme eesti häälega sai koostatud 18 eestikeelset lauset ning 36 kahekeelset lauset, millest kõik sisaldasid eesti keelt ning teise keelena esinesid inglise, prantsuse, hispaania, saksa ning vene keeled. Lisaks sünteesiti mudeliga ka lähteandmetes sisaldunud 22 eestikeelset lauset. Kuna varasemalt olemasoleva 9 keele näidislaused, mida T. Nekvinda jt [11] projektis ka demonstreeritud on, kõlasid valideerimisetapis autori hinnangul praktiliselt identselt, siis ei hakatud varasemate keelte vahelisi sünteese testima.

6.2 Hindamine ja tulemused

Koostatud lausetega[5] viidi läbi küsitlus 12 hindajaga, kellest enamik räägib eesti keelt emakeelena ning ülejäänud oskavad eesti keelt soravalt. Igale näitele andis hindaja loomulikkuse ning arusaadavuse hinde skaalal ühest viieni. Ära tuleb märkida, et hinnanguid andes ei olnud sünteesitud klippidel võrdluseks teiste lahenduste sünteese, vaid hindamine toimus hindaja subjektiivsel arvamusel ainult sünteesitud heli kohta. Hinnangute põhjal arvutati mudeli keskmised tulemushinnangud ehk *mean opinion score* eraldi iga hääle kohta mitmekeelsetele näidetele ja kõikidele näidetele, lisaks ka kõigi häälte keskmised kokku. Tulemused on kajastatud Tabelis 2.

Lähteandmetes sisaldunud näidetega hinnati sünteesitud lause arusaadavust enne ja pärast lause õige lindistuse kuulamist viie palli skaalal. Teadmata enne lauset, oli arusaadavus 2.9, peale lindistuse kuulamist oli sünteesitud lause arusaadavus aga 3.9. Märgatav vahe näitab, et mudel on õppinud ära küll häälte karakteristikud, kuid kvaliteetsest häälest jääb veel palju puudu. Hindajad tõid välja, et kuigi ilmekusest ja loomulikkusest saadi enamvähem aru, olid hääled summutatud ning vahel esines ootamatut sahinat, mistõttu oli ilma kontekstita raske sisust aru saada.

[5] Mõned näited on saadaval siin.

Hääl	*Mean opinion score* mitmekeelsetele näidetele	*Mean opinion score* kõigile näidetele
Hääl #1	2.76	2.94
Hääl #2	3.6	3.68
Hääl #3	2.26	2.34
Hääl #4	3.33	3.39
Hääl #5	3.00	2.93
Hääl #6	3.07	3.17
Hääl #7	2.95	3.02
Kõik hääled	3.00	3.08

Tabel 2. Häältele antud hinnangute keskmised.

Hinnangutest kajastus, et kui mitmekeelsetel näidetel oli loomulikkus umbes sama, siis arusaadavus oli halvem. Arusaadavust langetasid kõige rohkem just keele üleminekupiirkonnad, kus esines pause ja artefakte. Mudeli kvaliteeti tõmbasid osaliselt alla väikesed kirjavead, mis esinesid metafailides. Lindistustes esines vahepeal sõnalisi erinevusi metafailiga, näiteks esines ühes lindistuses sõna „sellega“, kuid metafailis oli samal kohal kirjas „seda“. Keeleliselt on tegu väga marginaalsete vigadega, mida ka esines hõredalt, kuid isegi sellised ebatäpsed teksti ja heli vastavused võivad muuta mudeli kaalusid piisavalt, et sünteesimisel ettearvamatud vead tekivad.

6.3 Võimalikud parandused

Eesti keele kvaliteedimäära tõstmiseks tuleks tegeleda veel rohkem andmete eeltöötlusega. Kui ülejäänud keelte põhjal on näha, et mudeli arhitektuur suudab kvaliteetsete andmete korral saavutada häid tulemusi, mida on kajastatud T. Nekvinda jt [11] teadustöös ning saavutatud ka käesolevas töös, siis autori hinnangul on eestikeelne süntees märksa madalama kvaliteediga. Näiteks saab korrigeerida metafailide vastavust lindistatud klippidele, muuta metafailides kõik arvud kirjalikuks. Andmete korrastamine ei pruugi olla aga ainuke vajalik täiendusviis kvaliteedi tõstmiseks, kuna ka T. Nekvinda jt [11] töös esines mitmekeelsetes lausetes vahel ebaharilikke pause. Kuna eestikeelseid andmeid oli teiste keelte andmetest tunduvalt rohkem ka peale andmete filtreerimist, siis töötas mudel teiste keelte andmeid mitu korda rohkem läbi. Seega võis ka andmestike mahtude mitte tasakaalus olemine olla eesti keele madalama kvaliteeditaseme põhjustajaks. Lisaks, kuna igal keelel on omad iseärasused, võib neid olla mõne keelte puhul raskem õppida.

7. Kokkuvõte

Kõnesüntees on tänapäeva maailmas kasvavalt nõudlik valdkond, samuti on tavalisemaks saamas keelte vaheldumine kõnes. Kui vanemates lahendustes põhines süntees inimese kõnesüsteemi modelleerimisele või manuaalselt seatud keelelistele tunnustele, siis viimastel aastatel on hakatud kasutama masinõpet, eelkõige tehisnärvivõrke, et asendada kõne sünteesimise komponente või isegi kõnesünteesi tervenisti. Närvivõrgud on valdkonnas saanud palju kajastust nende võrdlemisi vähese eeltöötluse vajaduse ja valmis mudeli kompaktsuse tõttu. Töös tutvuti osade uuemate töödega, kuhu on lisatud kõne sünteesijale ka paindlikkus võtta sisendiks mitut erinevat keelt korraga. Need lahendused töötavad enamasti eri keelte ühekeelsetel andmestikel, seega skaleerimine uutele keeltele on üsna lihtne. Ühelegi koostatud mitmekeelsele mudelile polnud veel aga kaasatud eesti keelt. Käesoleva töö raames valmis esimene eesti keelega töötav mitmekeelne tekstist kõne sünteesija. Sünteesija koostamiseks toetuti olemasolevale arhitektuurile, mille kasutatud keeltele lisati juurde eesti ja inglise keel. Töös on kirjeldatud sünteesija tegemise etapid. Kokku töötab valminud sünteesija 11 keelega, millest 9 varasemalt kasutatud keele sünteesikvaliteet on hinnanguliselt sama baaslahendusega. Koostatud eestikeelsete ja mitmekeelsete lausete põhjal viidi koos hindajate abiga läbi mudeli eesti keele kvaliteedi hindamine. Lõpuks toodi välja, mida edasi teha saaks, et saada mudel, mille eesti keele kvaliteet oleks veel parem.

8. Kasutatud allikad

[1] T. Tu, Y.-J. Chen, C.-c. Yeh and H.-y. Lee, “End-to-End Text-to-Speech for Low-Resource Languages by Cross-Lingual Transfer Learning,” in *Interspeech*, Graz, 2019.

[2] K. R. Chandu, S. K. Rallabandi, S. Sitaram and A. W. Black, “Speech Synthesis for Mixed-Language Navigation Instructions,” in *Interspeech*, Stockholm, 2017.

[3] Z. C. Lipton, J. Berkowitz and C. Elkan, “A Critical Review of Recurrent Neural Networks for Sequence Learning,” arXiv, 2015.

[4] I. Kuzovkin, R. Vicente, M. Petton, J.-P. Lachaux, M. Baciu, P. Kahane, S. Rheims, J. R. Vidal and J. Aru, “Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex,” *Commun Biol*, vol. 1, no. 107.

[5] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark and R. A. Saurous, “Tacotron: Towards End-to-End Speech Synthesis,” in *Interspeech*, Stockholm, 2017.

[6] R. E. Donovan, “Trainable Speech Synthesis,” 1996.

[7] R. A. Khan and J. S. Chitode, “Concatenative Speech Synthesis: A Review.,” in *International Journal of Computer Applications*, 2016.

[8] S. Lemmetty, *Review of Speech Synthesis Technology*, Helsinki University of Technology, 1999, pp. 28-50; 87-88.

[9] X. Tan, F. Soong and T.-Y. Liu, “A Survey on Neural Speech Synthesis,” arXiv, 2021.

[10] O. Karaali, G. Corrigan, I. Gerson and N. Massey, “Text-To-Speech Conversion with Neural Networks: A Recurrent TDNN Approach,” in *Eurospeech 1997*, Rhodes, 1998.

[11] T. Nekvinda and O. Dušek, “One Model, Many Languages: Meta-Learning for Multilingual Text-to-Speech,” in *Interspeech*, Online, 2020.

[12] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg and B. Ramabhadran, “Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning,” in *Interspeech*, Graz, 2019.

[13] G. Ruggiero, E. Zovato, L. Di Caro and V. Pollet, “Voice Cloning: a Multi-Speaker Text-to-Speech Synthesis Approach based on Transfer Learning,” arXiv, 2021.

[14] D. Griffin and J. Lim, “Signal estimation from modified short-time Fourier transform,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1983.

[15] A. van den Oord, S. Dieleman, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, *WaveNet: A Generative Model for Raw Audio*, arXiv, 2016.

[16] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman and K. Kavukcuoglu, “Efficient Neural Audio Synthesis,” in *35th International Conference on Machine Learning*, 2018.

[17] I. Sutskever, O. Vinyals and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” in *27th International Conference on Neural Information Processing Systems*, Montreal, 2014.

[18] D. Bahdanau, K. Cho and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *3rd International Conference on Learning Representations*, San Diego, 2015.

[19] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis and Y. Wu, “Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, 2018.

[20] H. Tachibana, K. Uenoyama and S. Aihara, “Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.

[21] A. A. Plantanios, M. Sachan, G. Neubig and T. Mitchell, “Contextual Parameter Generation for Universal Neural Machine Translation,” in *Empirical Methods in Natural Language Processing*, 2018.

[22] K. Park and T. Mulc, “CSS10: A Collection of Single Speaker Speech Datasets for 10 Languages,” Interspeech, 2019.

[23] L. Piits, *The Corpus of Speech Synthesis of the Institute of the Estonian Language,* 2016.

[24] E. Meister, *Corpus of Radio News,* 2014.

[25] M. Fišel and L. Rätsep, *Speech Corpus of Estonian News Sentences,* 2020.

[26] K. Ito and L. Johnson, “The LJ Speech Dataset,” 2017.

[27] University of Tartu, “UT Rocket,” share.neic.no.

I. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Rasmus Lellep,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose, Mitmekeelne kõnesüntees eesti keelega, mille juhendaja on Liisa Rätsep, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Rasmus Lellep

04.08.2021