TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Mona Tolmats

Automaatne kõnesünteesi kvaliteedi hindamine soome-ugri keeltele

Bakalaureusetöö (9 EAP)

Juhendaja:
Liisa Rätsep, MSc

Tartu 2025

Automaatne kõnesünteesi kvaliteedi hindamine soome-ugri keeltele

Lühikokkuvõte:

Sünteeskõne kvaliteedi automaatne hindamine kiirendab kõne sünteesivate mudelite arendust ja testimist, asendades standardse inimeste hinnangul põhineva keskmise arvamuse skoori leidmise. Antud tehnoloogia väljatöötamine on eriti oluline väiksema kõnelejaskonnaga keelte jaoks, mille puhul puuduvad ulatuslikud kõne- ja tekstiandmekogud ning piisava hulga hindajate kaasamine on keerulisem. Käesoleva töö eesmärk on treenida mudel, mis hindab eestikeelse sünteeskõne naturaalsust ja üldistub ka erinevatele soome-ugri keeltele. Treening- ja testandmeteks võeti nelja erineva aasta hindamiste tulemused. Mudelite üldistatavust hinnati võru keele põhjal. Esmalt treeniti wav2vec 2.0 mudel, seejärel peenhäälestati samal arhitektuuril põhinev mudel, mis oli eeltreenitud kasutades SCOREQ kaofunktsiooni. Viimaks peenhäälestati mudel UTMOSv2. Eksperimentide tulemused näitasid, et parima korrelatsiooni inimese poolt antud hinnangutega ja suurima üldistatavuse võru keelele saavutas UTMOSv2.

Võtmesõnad: automaatne kõnesünteesi hindamine, soome-ugri keeled, tehisnärvivõrk, wav2vec 2.0, UTMOSv2, SCOREQ kaofunktsioon

CERCS: P176 Tehisintellekt

Automatic Speech Synthesis Quality Assessment for Finno-Ugric Languages

Abstract:

Automatic evaluation of synthesised speech quality accelerates the development of text-to-speech models by replacing costly human listening tests based on mean opinion score. This capability is particularly valuable for low-resource languages, where only limited speech and text corpora are available and finding an adequate group of human evaluators is particularly challenging. The aim of this thesis is to train a model that evaluates the naturalness of Estonian synthetic speech and generalises to other Finno-Ugric languages. A wav2vec 2.0 was trained to predict mean opinion scores on Estonian text-to-speech models outputs. Separately, a wav2vec 2.0 model pre-trained using the SCOREQ loss function was fine-tuned, and the UTMOSv2 model was also adapted through fine-tuning. Training drew on three distinct datasets, while evaluation of cross-lingual generalisability was conducted on a single Võro-language test set. The experimental findings indicated that UTMOSv2 achieved the highest Pearson and Spearman correlations with human judgments and demonstrated superior generalisation to previously unseen Finno-Ugric languages.

Keywords: automatic speech synthesis evaluation, Finno-Ugric languages, artificial neural network, wav2vec 2.0, UTMOSv2, SCOREQ loss function

CERCS: P176 Artificial intelligence

Sisukord

Sissejuhatus

Loomuliku keele ja kõne töötluse valdkonna üks olulisemaid uurimissuundi on tekstipõhine kõnesüntees. Kõnesünteesiks nimetatakse inimkõne tehislikku genereerimist kasutades erinevaid arvutuslikke mudeleid ja algoritme. Alates 2010. aastast on sügavõppel põhinevate tehisnärvivõrkude kasutamine saanud sel alal valdavaks lähenemisviisiks [1]. Võrreldes varasemate konkateneerivate ja statistiliste mudelitega tagavad tehisnärvivõrgud kvaliteetsema ja loomulikuma inimkõne.

Sünteeskõne hindamiseks kasutatakse keskmise arvamuse skoori, mille puhul inimene annab viiepalliskaalal hinnangu. Kuigi antud meetod on standard, on tegemist ajamahuka ja kuluka hindamismeetodiga. Usaldusväärse tulemuse saavutamiseks tuleks kaasata vähemalt 30 hindajat [2]. Sealjuures ei tohiks iseseisvate hindamiste tulemusi omavahel otse võrrelda, kuna erinevad nii eksperimentaalsed tingimused kui ka hindajate individuaalsed eelistused.

Automaatne kõnesünteesi hindamine tagaks usaldusväärsema ja ühtsema tulemuse, kiirendades ühtlasi ka sünteeskõne mudelite arendusprotsessi. Antud tehnoloogia väljatöötamine on eriti oluline väiksema kõnelejaskonnaga keelte jaoks, millel puuduvad kõne- ja tekstiandmekogumid ning piisava hulga hindajate kaasamine on keerulisem. Sellist ühtset tehnoloogiat pole seni veel välja arendatud. Mitmete varem loodud hindamismudelite testimised on näidanud, et mudelid ei üldistu hästi tundmatutele, treeningprotsessis mitteesinenud sünteeskõne mudelitele. Eestikeelset sünteeskõnet hindavat mudelit varem loodud ei ole.

Käesoleva töö eesmärk on treenida eestikeelset sünteeskõnet hindav mudel, mis korreleerub inimese hinnangutega, ja uurida selle mudeli üldistusvõimet erinevatele soome-ugri keeltele. Mudelite treenimiseks kasutatakse eestikeelset sünteeskõnet kolmelt erinevalt hindamiselt ning üldistuvust teistele soome-ugri keeltele testitakse võrukeelsetel andmetel. Võrreldakse kolme lähenemise tulemusi, milleks on wav2vec 2.0[1] mudeli siirdeõpe, SCOREQ[2] kaofunktsioonil eeltreenitud wav2vec 2.0 mudeli ning UTMOSv2[3] mudeli peenhäälestamine.

Töö teoreetiline osa käsitleb esmalt närvivõrkude ja sügavõppe põhimõtteid ning seejärel automaatse kvaliteedihindamise meetodeid ja seniseid uurimusi. Praktilises osas kirjeldatakse

[1] https://docs.pytorch.org/audio/0.10.0/pipelines.html
[2] https://github.com/alessandroragano/scoreq
[3] https://github.com/sarulab-speech/UTMOSv2/tree/main

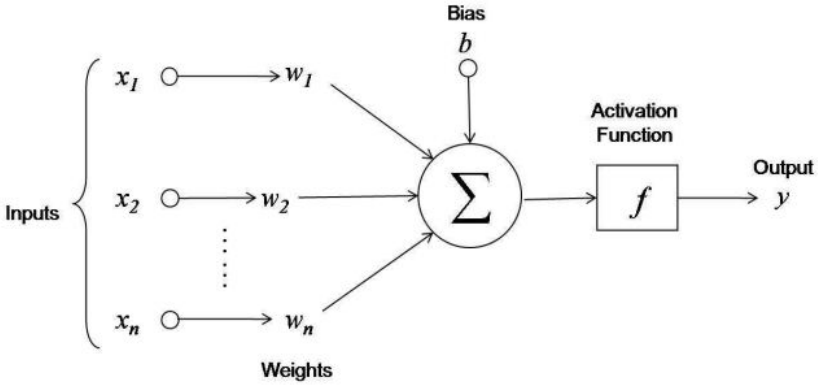
kasutatud andmestikke, läbiviidud eksperimente ning nende hindamismeetodeid. Viimaks esitatakse kolme mudeli tulemused, arutletakse nende usaldusväärsuse ja üldistusvõime üle ning pakutakse suundi edasiseks uurimuseks.

1. Töö teoreetiline taust

Peatükis antakse ülevaade töö teoreetilisest taustast. Keskendutakse tehisnärvivõrgu tööpõhimõtetele, kirjeldatakse transformeri ja konvolutsioonilise närvivõrgu arhitektuuri ning nende eripärasid. Antakse ülevaade tehisnärvivõrgu rakendusvõimalustest sünteeskõne analüüsimisel ja tutvustatakse hindamismeetodit „keskmise arvamuse skoor". Viimaks kirjeldatakse töö praktilises osas kasutatavate mudelite arhitektuure ja tööpõhimõtteid ning käsitletakse soome-ugri keeli.

1.1 Tehisnärvivõrk ja sügavõpe

Tehisnärvivõrk on masinõppe mudel, mille arhitektuuri loomisel on jäljendatud bioloogilise närvivõrgu struktuuri. Tehisnärvivõrk koosneb omavahel seotud üksustest ehk neuronitest, mis töötlevad sisendandmeid, õpivad mustreid ja seeläbi võimaldavad lahendada keerukaid ülesandeid [3]. Tegelikkuses on neuron funktsioon, mille tööpõhimõte on kujutatud joonisel 1. Neuroni sisendiks on n reaalarvust koosnev vektor, mille väärtused jäävad lõikku 0 kuni 1. Vektori elemendid korrutatakse läbi kaaludega, summeeritakse ja liidetakse nihe (*bias*). Saadud väärtusele rakendatakse aktivatsioonifunktsiooni, mille tulemuseks on neuroni lõppväljund, mis edastatakse järgmisele neuronile [4]. Aktivatsioonifunktsioon võimaldab mudelil õppida mittelineaarseid seoseid. Levinud aktivatsioonifunktsioonid on ReLU, sigmoid ja tahn [5].



Joonis 1. Neuron tehisnärvivõrgus [4]

Närvivõrgu struktuur on üles ehitatud neuronite kihiti paigutusele [3]. Iga kiht on justkui funktsioon, mis arvutab eelmise kihi väljundvektori põhjal uue vektori ning edastab selle järgmisele kihile. Kihtide arvu nimetatakse tehisnärvivõrgu sügavuseks. Sealt tuleb ka mõiste sügavõpe [3]. Närvivõrk koosneb minimaalselt ühest sisendkihist, kuid reeglina lisatakse veel

erinevaid peitkihte ja väljundkiht. Pärilevivõrk on üks kõige lihtsama arhitektuuriga närvivõrgu struktuure, kus andmed liiguvad vaid ühes suunas sisendkihist läbi peitkihtide väljundkihti [6].

Närvivõrgu treenimine juhendatud õppe puhul viiakse läbi mudeli sisenditest ja oodatavatest väljunditest koosneval andmestikul. Isejuhendatud õppel antakse mudelile vaid sisendandmed ning mudel õpib ise andmetest seoseid leidma. Treeningprotsess toimub iteratsioonide kaupa üle treeningandmestiku, ühte iteratsiooni kutsutakse epohhiks. Iga epohhi jooksul antakse mudelile ette plokkide ehk *batch*'ide kaupa terve sisendandmestik, mille põhjal mudel õpib ennustama väljundit [3]. Ploki lõpus hinnatakse ennustatud väärtuste ja tegelike väärtuste erinevust kaofunktsiooni abil ning iga epohhi lõpus leitakse kaofunktsiooni keskmine. Kaofunktsioon näitab ennustuste vastavust tegelikkusele [3]. Erinevaid kaofunktsioone valitakse vastavalt mudeli ülesandele. Regressioonimudeli puhul on üks standardfunktsioone keskmine ruutviga (*Mean squared error*, MSE), millega arvutatakse ennustatud ja tegelike väärtuste ruutude keskmine. Klassifitseerimismudeli puhul kasutatakse ristentroopia funktsiooni, mis mõõdab ennustatud tõenäosusjaotuse ja tegeliku jaotuse erinevust [7]. Mudeli treenimisel on eesmärk leida kaalud, mis minimeerivad kaofunktsiooni. Selleks kasutatakse gradientlaskumist, mis näitab, millises suunas kaale uuendada, et kaotus väheneks [8]. Kaale uuendatakse sammhaaval õpisammu ehk *leaning rate*'i alusel, mis määrab sammu pikkuse. Liiga väikse õpisammu korral võib kaofunktsioon jääda lokaalsetesse miinimumidesse, samas kui liiga suur õpisamm võib põhjustada kaotusefunktsiooni kõikumisi [3].

1.1.1 Vektoresitus

Närvivõrgule andmete esitamiseks tuleb need esmalt viia masinale arusaadavale kujule, milleks on reaalarvuline vektoresitus [9]. Antud vektoresitus moodustatakse mudeli esimeses kihis, kus igale unikaalsele elemendile määratakse unikaalne vektor. Vektori kaudu antakse mudelile informatsiooni erinevate tunnuste kohta, millest mudel hakkab hiljem õppima.

Närvivõrk töötleb korraga kindla arvu elemente. Olgu antud parameeter tähistatud x-ga ja selle levinud väärtused 16, 32, 128 või 256 [10]. Iga element, kas pilt, helifail või mõni muu sümbol, teisendatakse kindla pikkusega reaalarvulisteks vektoriteks. Olgu vektori mõõde tähistatud parameetriga y ja selle levinud väärtused 256, 512 või 1024 [11]. Moodustub $x \times y$-maatriks X, kus iga rida vastab ühele sisendelemendile. Sellist maatriksit saab kujutada järgmiselt:

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1y} \\ x_{21} & x_{22} & \dots & x_{2y} \\ \vdots & \vdots & \ddots & \vdots \\ x_{x1} & x_{x2} & \dots & x_{xy} \end{pmatrix},$$

kus x_{ij} tähistab i-nda elemendi j-nda tunnuse väärtust. Selline lähenemine võimaldab mudelil tõhusalt õppida sisendi struktuuri ja tähendust [10].

1.2 Transformer

Transformer on kindlat tüüpi järjendite teisendamisel (*sequence-to-sequence*) põhinev tehisnärvivõrk, mis pakuti esimest korda välja artiklis „Attention Is All You Need“ [10]. Transformer-arhitektuuri eripäraks võrreldes eelnevate tehisnärvivõrgu arhitektuuridega on enesetähelepanu mehhanism, mis võimaldab sisendi elemente vaadelda paralleelselt. See tähendab, et mudel on võimeline mõistma sisendi konteksti ja seoseid andmetes. Selline paralleeltöötlus muudab treenimise kiiremaks ja tõhusamaks kui varasemate lahenduste, näiteks rekurentsete RNN- ja LSTM-arhitektuuride, puhul [9]. Antud alapeatükk on kirjutatud eelmainitud artikli ning artikli „The Illustrated Transformer“ [12] põhjal.

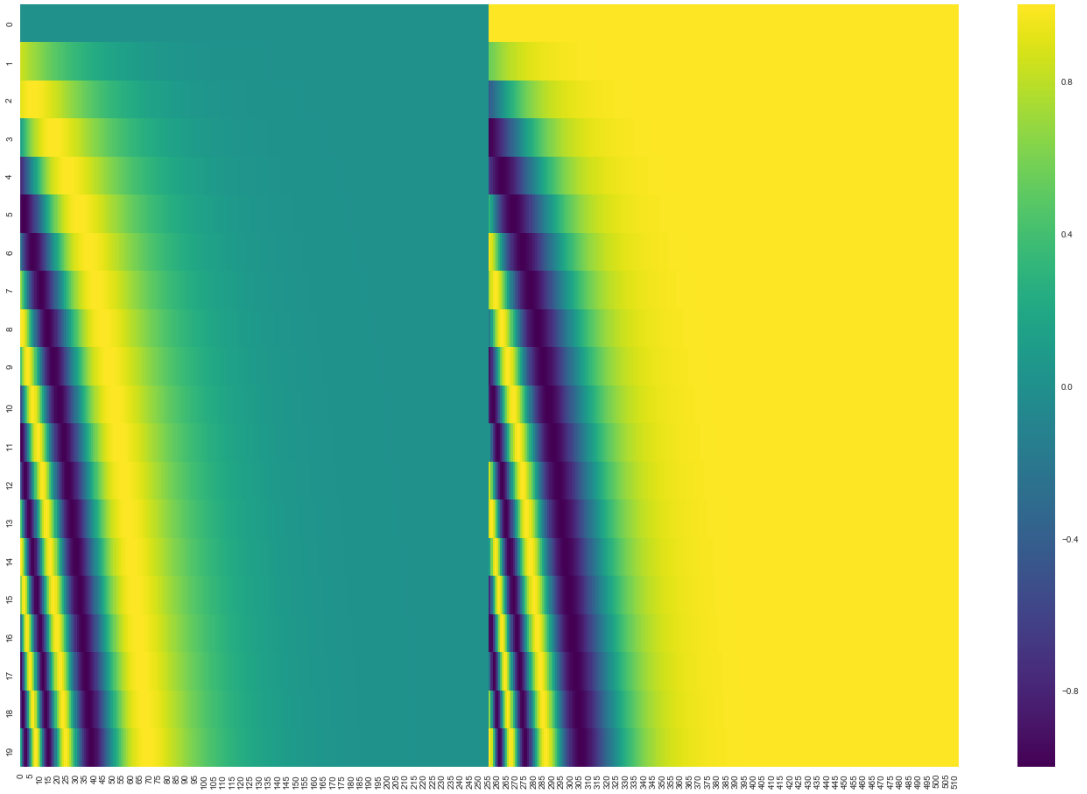
1.2.1 Positsiooni kodeerimine

Transformer ei kasuta rekurrentseid ega konvolutsioonilisi struktuure, mis säilitavad sisendelementide järjestuse loomulikult. Seetõttu tuleb elementide suhteline ja absoluutne asukoht sisendis kodeerida täiendavate vektorite abil. Transformer ise ilma lisakihita vaatleb kõiki elemente paralleelselt, mistõttu on kasutusele võetud positsiooni kodeerimine (*Positional Encoding*). Positsiooni kodeerimine on meetod järjestatud informatsiooni kodeerimiseks. Selleks lisatakse enne kooderisse ja dekooderisse sisestamist iga elemendi vektorile positsiooniteave, mis arvutatakse siinus- ja koosinusfunktsioonide abil:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right),$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right),$$

kus pos on elemendi indeks sisendjadas, i on vaadeldava elemendi indeks sisendjadas, mille kaugust määratakse, ja d_{model} on vektori mõõde. Arvutatava vektori mõõtmed peavad vastama elemendi vektori mõõtmetele, et neid oleks võimalik liita.



Joonis 2. Näide võimalikest positsiooni kodeerimise tulemusel saadud vektorite väärtustest [12]

Joonisel 2 on toodud näide, kus iga rida kujutab ühe kindla elemendi jaoks arvutatud kauguste vektorit. Vektori mõõde on 512 ja vektori väärtused jäävad lõikku -1 kuni 1. Jooniselt on näha, et madalamates mõõtmetes muutuvad positsiooni kodeerimisel arvutatud vektorite väärtused kiiremini, samas kui kõrgemate mõõtmete korral toimuvad muutused aeglasemalt. Selline lainepikkuse varieerumine võimaldab mudelil eristada nii lähestikku kui ka kaugemal paiknevaid sisendi positsioone, toetades seeläbi erinevate kaugusvahemike modelleerimist.

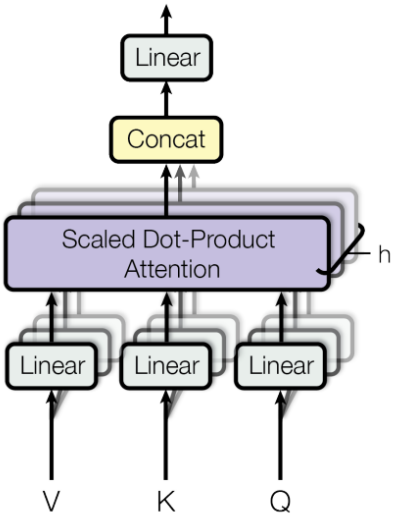
Siinus- ja koosinusfunktsioonide kasutamine ei ole ainuke võimalus, kuid Vaswani jt on maininud, et antud funktsioonid võimaldavad määrata elementide vahelisi kaugusi ka juhul, kui sisendi pikkus erineb treeningandmetes leiduvatest pikkustest. Põhjuseks on asjaolu, et siinus- ja koosinusfunktsioonid on perioodilised, seetõttu üldistub lahendus ka uutele sisendi pikkustele. Näiteks võib masintõlkes või tekstikokkuvõtete genereerimisel sisendjada olla kas lühem või pikem kui mudeli treeningandmetes nähtud jadad [13].

1.2.2 Enesetähelepanu mehhanism

Enesetähelepanu kihis luuakse iga elemendi vektori kohta kolm vektorit *query*, *key* ja *value*. Selleks korrutatakse vektor vastavalt kaalumaatriksitega W^Q, W^K ja W^V. Olgu maatriksite W^Q ja W^K dimensioon on d_k ning maatriksi W^V dimensioon d_v. Edasi arvutatakse, kui palju sõltub üks element teistest sisendjada elementidest. Selleks leitakse elemendi *query* vektori ja kõigi teiste elementide *key* vektorite skalaarkorrutised, mis jagatakse läbi parameetri y ruutjuurega, et liialt suuri väärtusi stabiliseerida ja muuta õppimine sujuvamaks. Edasi rakendatakse saadud väärtustele *softmax* funktsiooni, mille tulemusena jäävad kõik väärtused lõikku 0 kuni 1 ning nende summa on 1. Saadud väärtused näitavad, kui palju on konkreetsetel positsioonidel asuvad elemendid seotud algselt vaadeldud elemendiga. Viimaks korrutatakse väärtused *value* vektoriga ja summeeritakse. Kui eelnev mõttekäik lihtsustada maatriksite kujule, saadakse järgmine valem enesetähelepanu arvutamiseks:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V,$$

kus Q, K ja V on maatriksid, mis on saadud maatriksi X korrutamisel vastavalt W^Q, W^K ja W^V maatriksitega.



Joonis 3. Mitme pea enesetähelepanu mehhanism [10]

Tegelikkuses käib arvutamine maatriksite kujul ja viiakse läbi h arvutust ehk kasutatakse mitme pea enesetähelepanu mehhanismi. Joonisel 3 on näha, et protsessi teostatakse paralleelselt h korda, mis nõuab, et iga arvutuse jaoks on eraldi treenitud W^Q, W^K ja W^V kaalumaatriksid. Sellest tulenevalt tekib h maatriksit, mis tuleb omavahel ühendada. Saadud $x \times hy$-maatriks

korrutatakse varasemalt treenimise käigus saadud kaalumaatriksiga W^0. Tulemuseks on maatriks, mis on ka enesetähelepanu mehhanismi alamkihi väljund.

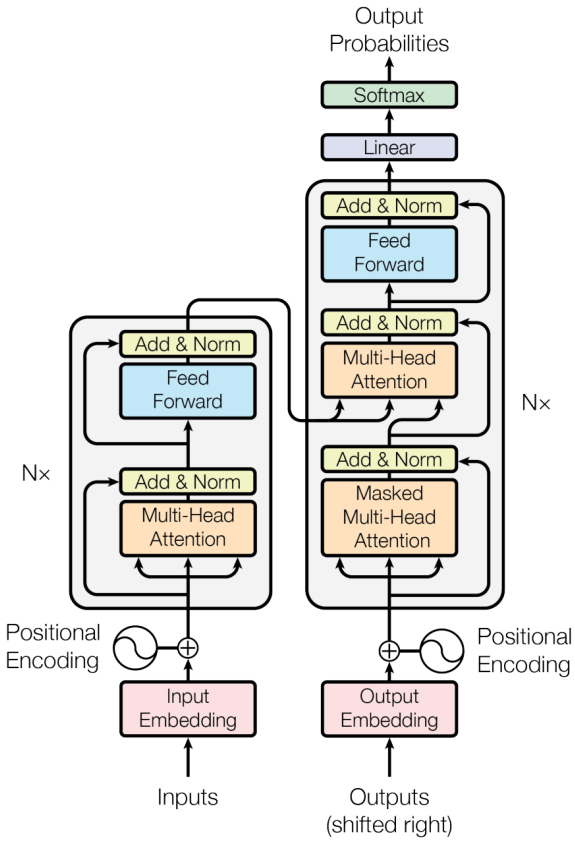
1.2.3 Transformer-mudeli arhitektuur

Esmakordselt välja pakutud transformeri mudelis kasutasid Vaswani jt [10] kooder-dekooder arhitektuuri. Hiljem on loodud ka transformereid, mis koosnevad ainult koodrist (BERT [14]) või dekoodrist (GPT [15]). Kooder teisendab sisendi tähenduslikeks tunnusvektoriteks, mille põhjal lahendatakse erinevaid analüüsi- ja klassifitseerimisülesandeid. BERT on loodud teksti analüüsimiseks. Dekooder genereerib aga uue väljundi, ennustades igal sammul järgmise elemendi, GPT puhul järgmise sõna, lähtudes eelnevast kontekstist.

Kooder on N närvivõrgu kihist koosnev struktuur, kus igal kihil on kaks alamkihti: mitme pea enesetähelepanu mehhanism ja kahe peitkihiga täissidus pärilevivõrk, kus iga kihi neuron on kaaluga ühendatud iga järgmise kihi neuroniga. Pärilevivõrku kirjeldab järgmine valem:

$$\mathrm{FFN}(x) = \max\left(0, xW_1 + b_1\right) W_2 + b_2,$$

kus x on sisend ja W_1, W_2, b_1, b_2 on parameetrid, mis defineeritakse vastavalt ülesandele. Dekooderile on lisatud veel teine mitme pea enesetähelepanu mehhanismi kiht, mille sisendiks on kooderi väjundvektorid. Kooderi ja dekooderi alamkihid kaale omavahel ei jaga. Transformeri kaalumaatriksid täidetakse treenimise algul juhuslikult. Üks võimalus on kasutada väikseid juhuslikke väärtusi mõne standardse jaotuse alusel.



Joonis 4. Transformeri arhitektuur [10]

Joonisel 4 on näha artiklis „Attention Is All You Need“ välja toodud transformeri arhitektuur. Enne kooderisses jõudmist teisendatakse andmed vektoresituseks ja rakendatakse positsiooni kodeerimise meetodit. Dekooderi väljund läbib lineaarse kihi, mis teisendab dekooderis õpitu elementideks, millele *softmax* annab tõenäosusjaotuse.

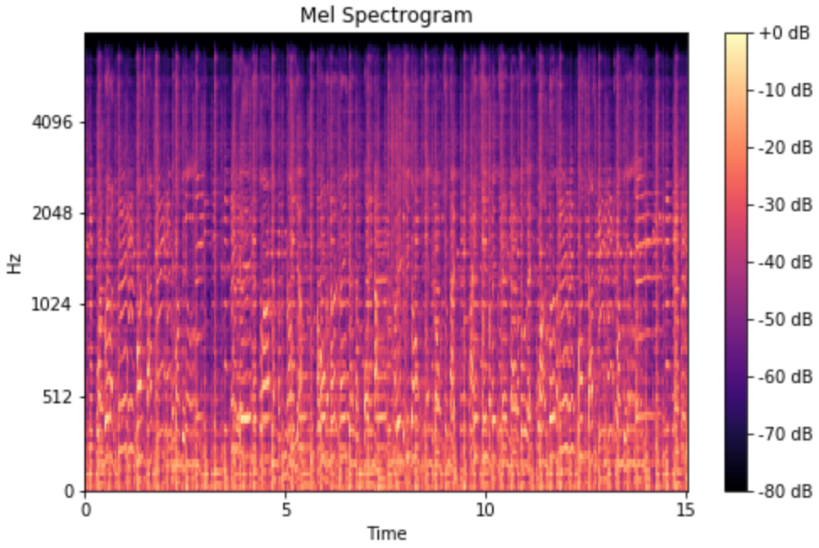
1.3 Konvolutsioonilised tehisnärvivõrgud

Konvolutsioon tähendab sama funktsiooni rakendamist erinevatele sisendi osadele. Konvolutsioonilised tehisnärvivõrgud on laialdaselt kasutuses just pilditöötluses, kus mudel otsib pildilt lokaalseid ruumilisi tunnuseid. Antud olukorras on tunnused defineeritud kolmemõõtmeliste maatriksite ehk tensoritena [9]. Konvolutsioonilised närvivõrgud koosnevad tavaliselt mitmest järjestikusest konvolutsioonikihist. Iga kiht rakendab sisendandmele kuni paarikümnest pikslist koosnevaid ruudukujulisi tensoreid filtri aktivatsiooni ehk esindatavuse leidmiseks [9]. Iga filtri skalaarkorrutis konkreetse pildi alaga loob tunnuskaardi, mis edastatakse järgmisele kihile, võimaldades hierarhilist õppimist. Esmalt analüüsitakse lihtsamaid tunnuseid ja seejärel kombineeritakse keerukamaid mustreid. Selline kihiline arhitektuur võimaldab

konvolutsioonilisel närvivõrgul tõhusalt õppida ja töödelda kõrge dimensiooniga visuaalseid andmeid [9].

1.4 Helisignaal ja akustilised tunnused

Üks helitöötluse keskseid probleeme on heli pidev muutumine ajas. Erinevalt tekstitöötlusest, kus teksti on võimalik jaotada lõplikuks hulgaks diskreetseteks väärtusteks (sõnad või nende alamosad), ei ole heli puhul nii sirgjoonelist lahendust [16]. Lisaks on toorheli töötlemine arvutuslikult kallis ja mitmel juhul ka ebaefektiivne, mistõttu kasutatakse mudeli treenimiseks eelnevalt väljaarvutatud akustilisi tunnuseid nagu mel-spektrogramm ning *mel-frequency cepstral coefficients* (MFCC) [17]. Teine levinud lahendus on treenida kooder, mis teisendab toorheli vektoriteks, mis esindavad 10-20ms pikkuseid helifragmente [16].



Joonis 5. Näide mel-spektrogrammist [18]

Spektrogrammil kujutatakse heli spektri muutust ajas, mis on reeglina saavutatud lühiajalise Fourier' teisenduse kaudu. Tegemist on 10-20ms helifragmentide lineaarse kujutamisega [19]. Selline lineaarskaalal põhinev spektrogramm ei kajasta aga inimese kuulmistaju olemust, kuna madalaid sagedusi eristab inimkõrv paremini. Seetõttu on kasutusele võetud mel-spektrogrammid, mille puhul kujutav vertikaaltelg teisendatakse mel-skaalaks [20]. Mel-skaala on loodud vastavalt inimkõrva sagedustajule. Alla 1000 Hz on sagedused jaotatud lineaarselt, kõrgematel sagedustel logaritmiliselt. Nii rõhutatakse mel-spektrogrammidel rohkem madalamaid sagedusi [18]. Joonisel 5 on näide mel-spektrogrammist. Närvivõrgule edastatakse mel-spektrogrammid vektoritena maatrikskujul, millele rakendatakse pilditöötlust. MFCC on edasiarendus mel-

spektrogrammist, kus mel-spektrogrammile rakendatakse logaritm- ja koosinusfunktsioone, et eraldada heli tunnusvektorid [21].

TTS mudelite väljundiks on reeglina heli tunnusvektorid, mis esindavad kõne akustilisi omadusi numbrilisel kujul, näiteks mel-spektrogrammid. Nende põhjal genereeritakse vokoodri abil lõplik helisignaal. Vokooder on mudeli spetsiaalne komponent, mis teisendab spektraalse kujutise lainekujuliseks helisignaaliks. Vokoodrid võivad olla reeglipõhised või kasutada sügavõppemeetodeid. Närvivõrkudel põhinevad mudelid loovad aga loomulikuma kõlaga sünteeskõnet [8].

1.5 Tehisnärvivõrkudel põhinev kõnesüntees

1.5.1 Sissejuhatus kõnesünteesi tehnoloogiasse

Tekstipõhise kõnesünteesi (*Text-to-speech*, TTS) eesmärk on teksti alusel genereerida loomulikult kõlav inimkõne. Antud tehnoloogia mängib olulist rolli masinatele kõnevõime andmisel ja on üks põhilisi uurimissuundi tehisintellekti ning loomuliku keele- ja kõnetöötluse valdkonnas [1]. Esimeste arvutipõhiste kõnesünteesi meetodite loomisel tugineti reeglipõhistele süsteemidele või kõnefragmentide kokkuliitmisel baseeruvatele ehk konkateneerivatele algoritmidele, mille puhul eelnevalt salvestatud foneemid, difoonid ja silbid valitakse välja ning liidetakse järjestikku [22]. Statistilise masinõppe arengu käigus kujunes välja mudeli tüüp (*Statistical Parametric Speech Synthesis*, SPSS), mille eesmärk on hinnata kõne akustiliste parameetrite (spekter, põhitoon ja kestus) tõenäosusjaotusi, et selle põhjal sünteesida inimkõne [23]. Alates 2010. aastatest on närvivõrkudel põhinev kõnesüntees järk-järgult muutunud valdavaks meetodiks, tagades märkimisväärselt parema häälekvaliteedi [1].

1.5.2 Keskmise arvamuse skoor

Keskmise arvamuse skoor (*Mean Opinion Score*, MOS) on standardne hindamismeetod, mida kasutatakse TTS mudelite sünteesitud kõne hindamiseks [8]. Hindamisprotsessis kuulavad inimesed erinevaid helifaile ja annavad hinnangu viiepalliskaalal (1 – ebaloomulik, 5 – väga loomulik ja naturaalne). Hinnangud kogutakse sõltumatult ehk iga sünteeskõne näidet hinnatakse eraldi, ilma viiteta algupärasele tekstile või muudele näidetele, et vältida süsteemide otsest võrdlust. Hinnangute keskmist loetaksegi konkreetse helifaili või mudeli MOS väärtuseks [24].

Siisiki leidub antud hindamismeetodi juures mitmeid sisulisi kitsaskohti. Wester jt [2] on läbi viinud Blizzard 2013 andmete analüüsi, millest järeldub, et MOS väärtuste tulemused muutuvad statistiliselt usaldusväärseks alles olukorras, kus on kaasatud vähemalt 30 hindajat.

Lisaks märgitakse, et erinevaid hindamisi ei tohiks omavahel otse võrrelda. MOS põhineb subjektiivsel hinnangul, mistõttu varieeruvad tulemused nii eksperimentaalsete tingimuste kui ka individuaalsete eelistuste tõttu. Wester jt [2] toovad välja, et MOS väärtust kasutatakse nii kõne kvaliteedi kui ka loomulikkuse hindamiseks. Antud artiklis tehtud analüüs näitab, et kui hindaja annab MOS väärtuse kvaliteedi põhjal, on see keskmiselt kõrgem, kui anda hinnang loomulikkuse põhjal. Sellegipoolest on MOS üks levinumaid sünteeskõne hindamismeetodeid [24].

1.6 Automaatne kõnesünteesi hindamine ja seotud tööd

Inimkuulajate abil kõnesünteesi kvaliteedi hindamine on küll standardne meetod, kuid muutub kiiresti ajakulukaks ja kalliks. Seda eriti kui hinnatavate mudelite arv kasvab. Automaatne kõnesünteesi hindamine võimaldaks kiiremat katsete kordamist ja suuremahuliste eksperimentide läbiviimist, mis kiirendaks omakorda kõnesünteesi tehnoloogiate arendamist [25]. Üks esimesi mudeleid, mis antud eesmärgil loodi, oli MOSNet [26]. Tegemist on konvolutsioonilisel tehisnärvivõrgul põhineva mudeliga sünteeskõne mel-spektrogrammi analüüsimiseks, et selle põhjal hinnata kõne naturaalsust ja loomulikkust. Antud mudel ei oma aga piisavat üldistusvõimet uute, treeningandmetes mitte esindatud TTS mudelite, hindamiseks [27].

Iga kõnesünteesi hindamistsükkel on erinev nii konteksti, hindajate kui ka juhiste poolest. Seetõttu ei üldistu automaatseks hindamiseks treenitud mudelid tihti uutele tundmatutele andmetele [27]. Selle valdava probleemi lahendamiseks on võetud kasutusele alusmudelid, mis on treenitud laiaulatuslike andmete peal täitma ühte konkreetset ülesannet. Siirdeõppe abil on võimalik neid mudeleid ümber õpetada väga erinevate eesmärkide täitmiseks, sealjuures on õppimine kiirem ja see on teostatav ka väiksema hulga andmetega [9].

2022. aastal avaldatud artikli tulemused näitavad, et mitmed erinevad isejuhendatud õppel põhinevad kõnemudelid Fairseq[4] projektist üldistuvad peale siirdeõpet ka tundmatutele andmetele [27]. Kõige paremaid tulemusi andsid konkreetses artiklis wav2vec 2.0 mudeli erinevad variatsioonid. Antud mudeli siirdeõppega eestikeelse sünteeskõne loomulikkuse hindamiseks tegeletakse töö praktilises osas.

[4] https://github.com/facebookresearch/fairseq

1.6.1 VoiceMOS Challenge

VoiceMOS Challange on esmakordselt 2022. aastal korraldatud rahvusvaheline võistlus, mille eesmärk on uurida ja edendada automaatsete kõnekvaliteedi hindamismeetodite arendamist, keskendudes sünteesitud või töödeldud kõne MOS väärtuse ennustamisele. Võistlejatele antakse standardiseeritud treening- ja testandmestikud erinevatest hindamistest, et tagada mudelite võrdlemine ühtsetel alustel [28]. Mudelite hindamisel arvestatakse keskmist ruutviga võrreldes inimese poolt antud MOS väärtustega. Samuti hinnatakse mitmesuguseid korrelatsioonikordajaid nii üksikute helifailide kui ka kogu mudeli tasandil, sealhulgas Spearmani järjestuskorrelatsiooni, mis mõõdab mudelite suutlikkust säilitada hinnangute suhtelist järjestust [29].

Igal aastal keskendutakse erinevatele alateemadele, mille raames osalejad mudeleid loovad. 2024. aasta võistluse esimeses arvestuses keskenduti kõrgekvaliteedilise sünteeskõne hindamisele. Selleks anti võistlejatel testandmestik üksnes parimate TTS mudelite ja häält moonutavate (*Voice conversion*) mudelite väljunditest, kus hinnatavad kvaliteedierinevused olid minimaalsed [28]. Ülesande raskus seisnes mudeli võimes eristada peeneid akustilisi erinevusi olukorras, kus kõigile näidetele oli omistatud kõrge MOS väärtus, mis nõudis mudelilt eriti täpset tundlikkust. Samal aastal saavutas UTMOSv2 mudel esikoha 7 arvestuses ning teise koha ülejäänud 9 arvestuses [30]. Käesoleva töö praktilises osas tegeletakse muuhulgas UTMOSv2 peenhäälestamisega eestikeelsele sünteeskõnele ning hinnatakse üldistusvõimet tundmatutele soome-ugri keeltele. Mudeli arhitektuuri ja kasutatavaid tehnilisi võtteid käsitletakse lähemalt alapeatükis 1.7.3.

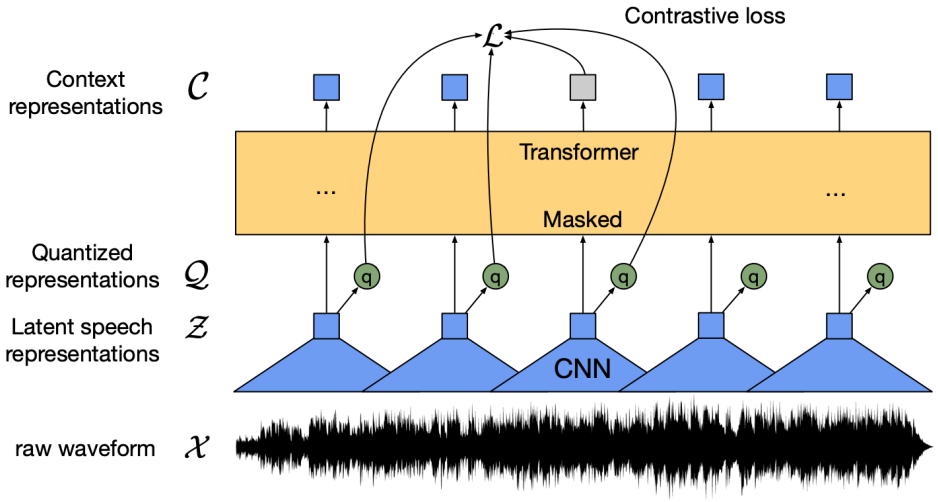
1.7 Sünteeskõne loomulikkust hindavad mudelid

Selles alapeatükis antakse ülevaade mudeli arhitektuuridest, mille siirdeõppe või peenhäälestamisega eestikeelsetele andmetele tegeletakse töö praktilises osas. Keskendutakse mudelite wav2vec 2.0 ja UTMOSv2 arhitektuuridele. Tutvustatakse kaofunktsiooni SCOREQ, mis on spetsiaalselt MOS väärtust ennustatavate mudelite jaoks loodud.

1.7.1 wav2vec2 2.0 mudeli arhitektuur

wav2vec 2.0 on Facebook AI poolt välja töötatud isejuhendatud õppel põhinev mudel, mis õpib otse toorsignaalist maskeerimise teel moodustama vektoreid, mis annavad mudelile informatsiooni nii helisignaali akustiliste kui ka prosoodiliste ja semantiliste tunnuste kohta [31]. Jooniselt 6 on näha, et mudel koosneb kolmest põhikomponendist. Esimene on konvolutsioonilisel närvivõrgul põhinev tunnuste kooder. Kooder koosneb mitmest konvolutsioonikihist, mille eesmärk on teisendada helisignaal lühikesteks latentseteks ehk peidetud vektoriteks, mis esindavad kõne madala taseme akustilisi tunnuseid antud ajahetkel.

Teiseks komponendiks on transformer-arhitektuuril põhinev närvivõrk, mis õpib järjestatud vektoritest enesetähelepanu mehhanismi abil helisignaali konteksti ehk informatsiooni üle kõigi vektorite. Väljundvektorid sisaldavad seega ka peale akustiliste tunnuste informatsiooni üle kogu helifaili. Erinevalt tavalisest transformerist, kus kasutatakse konteksti väljendamiseks staatilisi positsioonivektoreid, rakendab wav2vec 2.0 väikest konvolutsioonikihti, mis genereerib positsioonisõltuva paranduse igale vektorile. See annab mudelile relatiivse ajatunnetuse, võimaldades sujuvalt arvestada ajas paiknevate akustiliste tunnuste omavahelisi seoseid [31]. Kolmandaks komponendis viiakse pidevad latentsed vektorid kvantiseerimise teel lõplikuks hulgaks diskreetseteks väärtusteks, mis esindavad sünteeskõne fragmente [32].



Joonis 6. wav2vec 2.0 mudeli arhitektuur [31]

Sarnaselt keelemudelile BERT maskeeritakse wav2vec 2.0 õppimisalgoritmis tunnuste kooderis osa vektoreid, mis tähendab, et osa juhuslikult valitud vektoreid asendatakse kindla treenitud vektoriga. Mudel peab treeningprotsessi käigus õppima leidma konteksti põhjal maskeeritud vektorile õiget diskreetset väärtust kõigi võimalike väärtuste seast [32]. Tulemuseks on mudel, mis on õppinud kõne üldise esituse. Lisades juurde konkreetse eesmärgiga närvivõrgu kihte, mille sisendiks on wav2vec 2.0 väljundvektorid, saab siirdeõppe abil treenida mudelit täitma väga erinevaid ülesandeid võrdlemisi väikse treeningandmestikuga [31]. wav2vec 2.0 mudelit on treenitud nii kõnetuvastuseks [33], emotsioonide tuvastamiseks kõnest [34] kui ka MOS väärtuse ennustamiseks [27, 35]. Viimasele kahele artiklile tuginetakse ka töö praktilises osas, kui tegeletaksse wav2vec 2.0 siirdeõppega sünteeskõne hindamiseks.

1.7.2 SCOREQ kaofunktsioon

SCOREQ on 2025. aastal spetsiaalselt sünteeskõne MOS väärtust hindavate mudelite jaoks loodud kaofunktsioon [36]. Võrreldes MSE kaofunktsiooniga, pöörab SCOREQ tähelepanu MOS väärtuste omavahelisele järjekorrale, mis on oluline antud regressiooniülesande juures. Iga treeningsammu jooksul vaadatakse hinnatavale helifailile lisaks veel ühte sarnase MOS väärtusega ja ühte erineva MOS väärtusega helifaili [36]. Järgnevalt on kirjeldatud kaofunktsiooni arvutamise käik.

Olgu N hulk sünteeskõne näiteid koos MOS väärtusega, mida mudelile korraga ette söödetakse. Iga näide (x_i, y_i) sisaldab helifaili x_i ja sellele vastavat MOS väärtust y_i. Olgu konvolutsiooniline kooder tähistatud funktsiooniga $g : X \to H$, mis teisendab helifaili x_i latentseks vektoriks $h_i = g(x_i)$. Olgu järgnev lineaarne projektsioonikiht tähistatud funktsiooniga $f : H \to Z$, mis teisendab latentse vektori madaladimensiooniliseks vektoresituseks $z_i = f(h_i) \in \mathbb{R}^d$. Järgnevalt vaadatakse läbi kõik võimalikud kolmikud üle hulga N ning iga kolmiku kohta määratakse järgmine väärtus:

$$\mathcal{M}(i,j,k) = \begin{cases} 1, & \text{kui } |y_i - y_j| < |y_i - y_k| \\ 0, & \text{muul juhul} \end{cases},$$

kus kolmiku väärtus on 1, kui tingimus $|y_i - y_j| < |y_i - y_k|$ kehtib ja 0, kui ei kehti. Lisaks fikseeritakse iga kolmiku kohta ka konstant:

$$m_{i,j,k} = \frac{|y_i - y_j| - |y_i - y_k|}{N - 1}, \tag{1}$$

mis näitab vaadeldavate MOS väärtuste omavahelist paiknemist skaalal. Viimaks arvutatakse kaofunktsioon $\mathcal{L}$ üle hulga N:

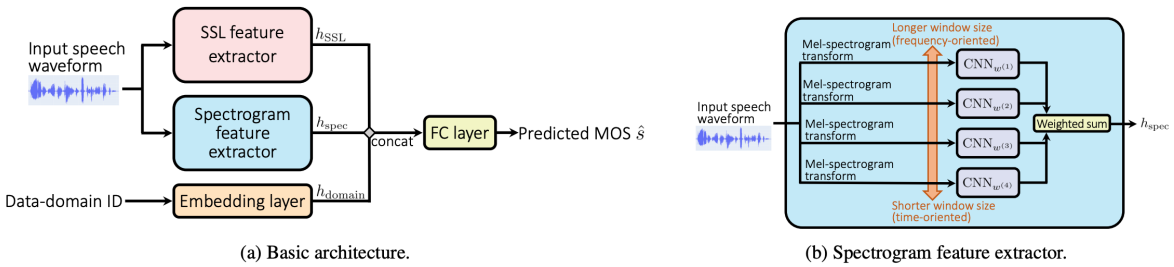
$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} \mathcal{M}(i,j,k) \Big(& \left\| f\big(g(\boldsymbol{x}_i)\big) - f\big(g(\boldsymbol{x}_j)\big) \right\|_2 \\ & - \left\| f\big(g(\boldsymbol{x}_i)\big) - f\big(g(\boldsymbol{x}_k)\big) \right\|_2 + \ m_{i,j,k} \Big), \end{aligned} \tag{2}$$

kus $\left\| f\big(g(\boldsymbol{x}_i)\big) - f\big(g(\boldsymbol{x}_j)\big) \right\|_2$ tähistab vektoresituste vahe vektori pikkuse ruudu leidmist. Leitud kaofunktsiooni väärtuse põhjal uuendatakse treeningprotsessi käigus mudeli kaale [36]. Artiklis treeniti wav2vec 2.0 mudelit SCOREQ kaofunktsioonil nii, et külmutati kooderi kaalud ja treeniti

vaid transformeri kaale. Lisati lineaarkiht, mis projitseerib transformeri väljundid ühemõõtmeliseks MOS ennustuseks [36]. Sellist lähenemist kasutatakse ka antud töö praktilises osas.

1.7.3 UTMOSv2 mudeli arhitektuur

UTMOSv2 on 2024. aastal VoiceMOS Challenge'i raames välja töötatud ja avaldatud mudel sünteeskõne loomulikkuse hindamiseks [30]. Jooniselt 7 on näha, et mudel koosneb kolmest komponendist ja neid ühendavast regressioonikihist. Esimeseks komponendiks on konvolutsioonilisel tehisnärvivõrgul põhinev EfficientNetV2 [37], mis on eeltreenitud ImageNet[5] andmestikul pildi tunnusvektori arvutamiseks [30]. Mudeli sisendiks on helisignaal, mille põhjal arvutatakse mel-spektrogrammid. Antud ülesande puhul arvutatakse mitme mel-spektrogrammi põhjal fikseeritud dimensiooniga tunnusvektor, mille abil antakse edasi kõne akustilisi tunnuseid. Mel-spektrogramme luuakse helifaili erinevatel ajahetkedel erineva pikkusega ajavahemike kohta. Pikemate ajavahemike puhul pöörab mudel rohkem tähelepanu heli sagedusele, väiksema ajaakna puhul aga heli muutumisele ajas. Teise komponendina kasutatakse wav2vec 2.0 mudelit, mis teisendab sisendiks oleva helisignaali tunnusvektoriks, kirjeldades nii akustilisi kui ka semantilisi kõne tunnuseid [30]. Kolmandas komponendis antakse igale andmestikule, mida treeningu jooksul kasutatakse, unikaalne tunnusvektor. Üheks andmestikuks loetakse siin ühe hindamise käigus kogutud MOS väärtuste hulka, kus olid samad hindamistingimused ja hindajad. Loodud tunnusvektor võimaldab mudelil tõhusamalt arvestada erinevatest hindamiskeskkondadest tulenevaid eripärasid ja saavutada parem üldistusvõime erinevate hindamiste vahel [30]. Viimaks kombineeritakse kolme komponendi tunnusvektorid kokku ja edastatakse regressioonikihile, mille väljundiks on ennustatud MOS väärtus.



(a) Basic architecture. (b) Spectrogram feature extractor.

Joonis 7. UTMOSv2 arhitektuur [30]

[5] https://image-net.org/index.php

UTMOSv2 kõigi treeningetappide jooksul kasutatakse kaofunktsiooni, mis arvestab nii MSE kui ka korrelatsiooni koefitsiendiga $\mathcal{L}_{\text{con}}$ [30]. Kaofunktsiooni valem matemaatilisel kujul on järgmine:

$$\mathcal{L}(s, \hat{s}) = \lambda_{\text{con}} \mathcal{L}_{\text{con}}(s, \hat{s}) + \lambda_{\text{mse}} \mathcal{L}_{\text{mse}}(s, \hat{s}), \tag{3}$$

kus $\mathcal{L}_{\text{con}}$ arvutatatakse järgneva valemiga:

$$\mathcal{L}_{\text{con}}(s, \hat{s}) = \sum_{i \neq j} \max\left(0, |(s_i - s_j) - (\hat{s}_i - \hat{s}_j)| - \alpha\right).$$

Antud valemites on s tegelike MOS väärtuste ja $\hat{s}$ ennustatud MOS väärtuste hulk. Parameetrid λ_{con}, λ_{mse} ja α olid vastavalt 0.2, 0.7 ning 0.2. Kui $\alpha = 0.2$, siis mudel ignoreerib viga, mis on väiksem kui 0.2 [30]. Antud kaofunktsiooni kasutatakse ka käesoleva töö praktilises osas UTMOSv2 mudeli peenhäälestamisel.

1.8 Hindamismeetodid

Lisaks keskmisele ruutveale (MSE), mis on toodud valemis (4), arvestatakse mitmes varasemas artiklis MOS väärtust ennustavate mudelite täpsuse hindamisel ka erinevate korrelatsioonikordajatega [27, 30].

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left(s_i - \hat{s}_i\right)^2 \tag{4}$$

Nendeks on lineaarne korrelatsioonikordaja (*Linear correlation coefficient*, LCC), Spearmani korrelatsioonikordaja (*Spearman's rank correlation coefficient*, SRCC) ja Kendalli korrelatsioonikordaja (*Kendall rank correlation coefficient*, KTAU). Järgnevalt on toodud korrelatsioonikordajate valemid, kus s on tegelike MOS väärtuste ja $\hat{s}$ ennustatud MOS väärtuste hulk, N antud hulkade suurus, rank järjestusfunktsioon ja sgn siinusfunktsioon. LCC matemaatiline kuju on näidatud valemis (5) [38]. Kordaja väärtused jäävad lõikku -1 kuni 1. LCC mõõdab ennustatud MOS väärtuse ja tegeliku MOS väärtuse lineaarse seose tugevust. Positiivse väärtuse korras on tegemist kasvava seosega, negatiivse väärtuse korral kahanevaga.

$$LCC(s,\hat{s}) = \frac{\sum_{i=1}^{N}\left(s_i - \bar{s}\right)\left(\hat{s}_i - \bar{\hat{s}}\right)}{\sqrt{\sum_{i=1}^{N}\left(s_i - \bar{s}\right)^2 \sum_{i=1}^{N}\left(\hat{s}_i - \bar{\hat{s}}\right)^2}} \tag{5}$$

$$\bar{s} = \frac{1}{N}\sum_{i=1}^{N} s_i \qquad \bar{\hat{s}} = \frac{1}{N}\sum_{i=1}^{N} \hat{s}_i.$$

SRCC matemaatiline kuju on näidatud valemis (6), mis mõõdab kahe arvuhulga omavahelise järjestuse seost [39]. Antud kordaja väljaarvutamisel kasutatakse järjestusfunktsiooni ehk järjekorranumbreid väärtuste järjestatud reas. Seega antud korrelatsioonikordaja arvutamisel otseselt MOS väärtust ei kasutata. SRCC jääb lõikku -1 kuni 1.

$$SRCC(s,\hat{s}) = 1 - \frac{6\sum_{i=1}^{N}\left(\text{rank}(s_i) - \text{rank}(\hat{s}_i)\right)^2}{N\left(N^2 - 1\right)} \tag{6}$$

KTAU jääb lõikku -1 kuni 1. Mida suurem on absoluutväärtus, seda tugevam on mudeli ennustuste ja tegelike MOS väärtuste järjestuse kokkulangevus [40]. KTAU arvutamisel kasutatakse samasuunaliste ja vastassuunaliste paaride suhtelist sagedust. KTAU matemaatiline kuju on näidatud järgmises valemis:

$$KTAU(s,\hat{s}) = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} \text{sgn}\left(s_i - s_j\right) \text{sgn}\left(\hat{s}_i - \hat{s}_j\right),$$

$$\text{sgn}(x) = \begin{cases} +1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}.$$

Eelmainitud korrelatsioonikordajaid kasutatakse ka antud töö praktilises osas arendatavate mudelite hindamisel ja võrdlemisel.

1.9 Soome-ugri keeled

Soome-ugri keeled moodustavad osa suuremast Uurali keelkonnast ja on tuntud oma mitmekesisuse ning murdeliste erinevuste poolest. Neid keeli kõnelevad rahvad ja kogukonnad on hajutatud üle Euroopa põhja-, ida- ja keskosa ning Aasia erinevate osade, mistõttu esinevad neil nii geograafilised kui ka ajaloolised murdeliigid. Läänemere soome-ugri keelte hulka kuuluvad eesti, soome ja karjala keel ning muud väiksemad keeled, millel on oma piirkondlikud murded ja dialektid [41, 42].

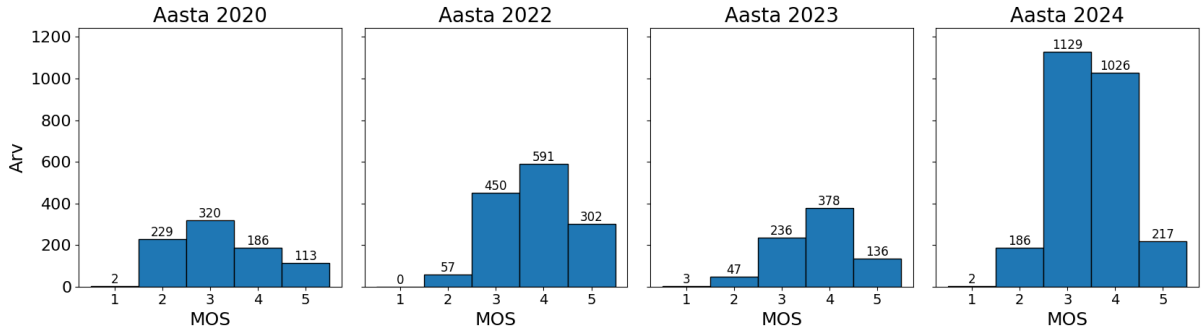
Üks läänemeresoome keeli on ka võru keel, mida räägitakse Kagu-Eestis [43]. Statistikaameti 2021. aastal korraldatud rahvaloenduse käigus selgus, et Eestis on võru keelt[6] oskavaid inimesi ligikaudu 100 000 ja eesti keele[7] kõnelejaid üle 1,2 miljoni inimese. Lisaks leidub veel väga väikse kõnelejaskonnaga läänemeresoome keeli, mida räägib alla paarisaja inimese. Nendeks on vadja, liivi ja isuri keeled [43].

[6] https://andmed.stat.ee/et/stat/rahvaloendus__rel2021__rahvastiku-demograafilised-ja-etno-kultuurilised-naitajad__voorkeeleoskus-murded/RL214462/table/tableViewLayout2
[7] https://andmed.stat.ee/et/stat/rahvaloendus__rel2021__rahvastiku-demograafilised-ja-etno-kultuurilised-naitajad__rahvus-emakeel/RL21442/table/tableViewLayout2

2. Andmed

Antud töös kasutati mudelite treenimiseks ja testimiseks nelja erineva projekti raames loodud eesti- ja võrukeelseid kõneandmestikke, mis sisaldavad nii inim- kui ka sünteeskõne. Kõigil andmestikel on viidud läbi hindamised MOS väärtuste leidmiseks. Treeningandmestikud pärinevad Rätsepa jt [44–46] avaldatud artiklitest aastatest 2020, 2022 ja 2024. Mudelite testimiseks kasutati 2023. aastal avaldatud artikli raames loodud võrukeelset kõneandmestikku, et hinnata mudeli üldistusvõimet ka mudelile seni tundmatutele soome-ugri keelele [47].

Kõneandmestike MOS väärtuste jaotus on välja toodud joonisel 8. Jooniselt on näha, et andmestik on tugevalt kallutatud keskmise ja kõrgema hinnangu pole ning väga madala hindega näiteid praktiliselt ei leidu. Sellest tulenevalt võib mudelil olla keerulisem MOS väärtuse alusel teha vahet heal ja väga heal TTS mudelil. Andmete jaotust ei muudetud ega tasakaalustatud, sest eesmärk oli säilitada reaalset hindajate jaotust ja testida mudelite võimet õppida tingimustes, mis vastavad reaalelulistele olukordadele.



Joonis 8. Eesti- ja võrukeelse sünteeskõne MOS väärtuste jaotus (vahemikes 0.5-1.5, 1.5-2.5, 2.5-3.5, 3.5-4.5, 4.5-5.5) neljas erineva aasta andmestikus. Tulpade kohal olev arv tähistab vastava vahemiku hinnete arvu.

Tabelis 1 on esitatud kõnesünteesi mudelite arv aastate lõikes. Arvestatud on ka inimkõnega, mida loetakse eraldi mudeliks. Hindamiste arv kajastab erinevatele helifailidele antud MOS väärtuste koguarvu. Ära on märgitud ka vastavate aastate madalaim ja kõrgeim hinne. MOS väärtused on arvutatud mitme hindaja hinnangute keskmise põhjal, mistõttu esinevad andmestikus ka murdarvud. Kokku oli eestikeelseid helifaile 4810 ning võrukeelseid 800. Mudeli treenimiseks kasutati 90% ja valideerimiseks 10% eestikeelsetest andmetest, vastavalt 4329 ja 481 helifaili. Andmed on jaotatud propotsionaalselt võrdselt nii MOS väärtuse kui ka erinevate aastate andmestikesse kuulumise järgi. Testimiseks kasutati võrukeelset andmestikku.

Tabel 1. Eesti- ja võrukeelsete andmestike hindamisstatistika aastate lõikes.

Aasta	Mudelite arv	Helifailide arv	Hindamiste arv	Madalaim MOS	Kõrgeim MOS
2020	5	850	17 000	1.4	5
2022	7	1400	5600	1.75	5
2023	8	800	2600	1	5
2024	9	2 560	12 800	1.4	5

Hindamiste tulemused olid algselt hajutatud mitmesse andmestikku ja küsitluse vastuste XML-failidesse. Ühtse suure andmestiku koostamiseks koondati esmalt küsimustike vastused ning leiti igale helifailile antud MOS väärtuste keskmine. Viimaks normaliseeriti kõigi sünteeskõne helifailide diskreetimissagedus 16 kHz-ni, et need vastaksid mudelite sisendite tingimustele.

2.1 2020. aasta andmestik

2020. aasta andmestikus on esindatud 5 mudelit, millega on genereeritud 850 eestikeelset helifaili, millele on kokku antud 17 000 hinnangut. Peamiseks TTS mudeliks antud andmestikus oli Tartu Ülikooli ja Eesti Keele Instituudi (EKI) koostöös eesti keelele kohandatud Deep Voice 3 tarkvara [48], mis põhineb tehisnärvivõrkudel ja võimaldab mitmehäälse kõnesünteesi mudeli treenimist. Kasutusele võeti erinevad sünteeshääled, mis treeniti erinevate treeningandmete alamhulkade peal. Lisaks Deep Voice 3 mudelitele ja EKI treeningkorpuse näidistele, mis esindasid inimhäält, hinnati ka EKI loodud HTS sünteeshääli[8] ja Google'i tõlkerakenduses[9] kasutatavat kõnesünteesi. Mudelite hindamine viidi läbi 20 tudengi seas. Hindajad kuulasid juhuslikult järjestatud 50 helifaili plokke, millele paluti anda hinnang 0.5 punkti täpsusega [44].

2.2 2022. aasta andmestik

2022. aasta andmestikus on esindatud 7 mudelit, millega on genereeritud 1400 eestikeelset helifaili, millele on kokku antud 5600 hinnangut. 2022. aasta andmestikus on TTS mudelite arhitektuuriks võetud FastPitch [49], mis võimaldab samuti mitmehäälset kõnesünteesi. Tegemist on mitteautoregressiivse transformer-arhitektuuril põhineva mudeliga, kus väljund genereeritakse paralleelselt, mitte järjestikku [50]. Mudeleid treeniti erinevate omadustega andmestike peal, milleks olid nii spetsiaalselt sisse loetud uudiste artiklid kui ka audioraamatute korpus. Eesmärk oli uurida, kui hästi suudavad transformeril põhinevad mudelid sünteesida eestikeelset kõne

[8] https://arhiiv.eki.ee/heli/
[9] https://translate.google.com/

kasutades erinevaid andmehulki ja andmetöötlusvõtteid. Kasutati ka erinevaid sünteeshääli. Mudelite hindamiseks koostati 24 küsimustikku, igaüks koosnes 115 sünteeskõne näitest. Iga küsimustikku hindas 5 inimest [45].

2.3 2023. aasta andmestik

2023. aasta andmestikus on esindatud 8 erinevat mudelit, millega on loodud 800 võrukeelset helifaili ja mida on kokku hinnatud 2600 korda. Mudelite treenimiseks kasutati nii eesti- kui ka võrukeelseid andmeid erinevatest andmestikest [47]. Eesmärk oli välja töötada mudel, mis on võimeline sünteesima võrukeelset kõne. Treeniti mitteautoregressiivset transformeri arhitektuuril põhinevat TTS mudelit, mis sarnanes FastPitch mudelile, ning 2022. aastal loodud mitmehäälset eestikeelset mudelit [45]. Kasutades erinevaid andmestikke ja siirdeõpet loodi 6 mudelit, millele lisati juurde veel inimekõne ning vokooderil töödeldud inimkõne näiteid. Hindamiseks valiti testkomplektist 100 juhuslikku lauset iga mudeli kohta ja genereeriti kõne kasutades erinevaid vokoodereid. Sünteeskõne hindasid 41 võru keelt kõnelevat inimest [47].

2.4 2024. aasta andmestik

2024. aasta andmestikus on esitatud 9 erinevat mudeli variatsiooni, millega on loodud 2560 helifaili ja mida on kokku hinnatud 12800 korda. Rätsepa jt [46] kirjutatud artikli eesmärk oli luua eestikeelne sünteeskõne mudel, mis suudab genereerida nii spontaanset vestluskõne kui ka etteloetavat monotoonsemat kõne. Treenimisel kasutati erinevaid sisseloetud ja spontaanse kõne andmestikke nii eesti kui ka võru keeles. Jällegi treeniti transformer-arhitektuuril põhinevat FastPitch mudelit. Hindamiseks loodi 24 küsitlust, mis sisaldasid 115 eestikeelset sünteeskõne näidet, neist igale küsitlusele vastas 5 inimest [46].

3. Eksperimendid

Selles peatükis antakse ülevaade käesolevas töös läbi viidud eksperimentidest eestikeelse sünteeskõne hindamismudeli loomiseks, mis üldistuks ka teistele soome-ugri keeletele. Esmalt treeniti siirdeõppe käigus wav2vec 2.0 mudelit hindama eestikeelset sünteeskõne, lisades mudelile juurde MOS väärtust ennustavad närvivõrgu kihid. Teise mudelina peenhäälestati wav2vec 2.0 arhitektuuril põhinev mudel, mis oli eeltreenitud hindama sünteeskõne kasutades SCOREQ kaofunktsiooni. Viimaks peenhäälestati UTMOSv2. Eksperimentide koodi väljatöötamisel kasutati ChatGPT keelemudeli[10] abi. Kõik eksperimendid viidi läbi Tartu Ülikooli teadusarvutuste keskuse klastris Rocket kasutades Tesla A100 40GB GPU ressursse [51].

3.1 wav2vec treenimine

Antud eksperimendi raames kasutati 95 miljoni treenitava parameetriga wav2vec 2.0 baasmudelit, mis on eeltreenitud 960 tunnil inglise keelsetel kõneandmetel. Mudeli ja arhitektuuri importimiseks kasutati `torchaudio.pipelines` Pyhtoni paketti[11]. Mudeli konvolutsioonilise tunnuste kooderi kaalud külmutati ja lisati MOS väärtuse ennustamiseks regressioonivõrk. Kasutati kahte lineaarkihti, mille vahel rakendati lineaarset normaliseerimisfunktsiooni, ReLU aktivatsioonifunktsiooni ja *dropout*'i tõenäosusega 0.2. See tähendab, et 20% neuronite kaaludest ei arvestata, et mudel õpiks kõne üldisemaid tunnuseid [52]. wav2vec 2.0 vektorite edastamisel MOS väärtust ennustavatele kihtidele rakendati keskmistamist (*mean pooling*) ehk arvutati iga ajahetke kirjeldava tunnusvektori elementide keskmine ja saadud tulemustest moodustati uus 768-elemendiline vektor, mis kirjeldab tervet audiofaili.

Mudeli treenimisel oli ploki suurus 32 ja kasutati AdamW optimeerijat õpisammuga 0.01. Kaofunktsiooniks valiti keskmine ruutviga ning õpisammu kohandati vastavalt selle tulemusele ReduceLROnPlateau skeemiga (*factor* 0,9, *patience* 10). Iga epohhi lõpus arvutati treening- ja valideerimiskaotused ning valideerimisel lisaks korrelatsioonikordajad LCC, SRCC ja KTAU, et jälgida treeningprotsessi. Treenimine lõpetati peale 30. epohhi, kui viimase 10 epohhi jooksul ei olnud valideerimisandmetel keskmine ruutviga paranenud. Salvestati madalaima valideerimisandmete keskmise ruutveaga epohhi kaalud. Mudeli üldistuvust teistele soome-ugri keeltele testiti võrukeelsetel andmetel.

[10] https://chatgpt.com/
[11] https://docs.pytorch.org/audio/0.10.0/pipelines.html

3.2 wav2vec 2.0 peenhäälestamine kasutades SCOREQ kaofunktsiooni

Antud eksperimendi raames kasutati SCOREQ kaofunktsioonil eeltreenitud wav2vec 2.0 mudelit, mille treenimisel kasutati erinevaid inglise keelseid kõneandmestikke koos sünteeskõne hinnangutega. Mudeli kaalud ja arhitektuur imporditi kasutades Pythoni paketti scoreq[12]. Mudelit treeniti sarnaselt eksperimendile SCOREQ kaofunktsiooni puudutavas artiklis [36]. Treeningprotsess viidi läbi kahes osas. Esmalt külmutati wav2vec 2.0 konvolutsioonilise tunnuste kooderi kaalud. Ülejäänud transformeri kooderi ja kvaniseerimiskihi kaale treeniti eestikeelsete andmete peal. Kasutati AdamW optimeerijat õpisammuga 1×10^{-5} ja õpisammu ajastajat OneCycleLR. Kaofunktsioonina kasutati SCOREQ funktsiooni (valem 2), kus konstant $m_{i,j,k}$ (valem 1) korrutati läbi muutujaga α, mis arvutati järgmise valemi alusel:

$$\alpha = 1 + \frac{2 \cdot \text{epohh}}{100}, \tag{7}$$

kus epohh tähistab vastavat treeningiteratsiooni. Mudelit treeniti algselt ilma α arvutamiseta, kuid siis jäi mudeli õppimine pidama SRCC väärtusel ≈ 0.5, mistõttu katsetati muutuja osakaalu suurendamisega. Valideerimisandmetel hinnati mudeli ennustatud vektoresituste omavahelist korrelatsiooni SRCC alusel. Salvestati kaalud, mille puhul SRCC oli kõige suurem. Treening lõpetati 27. epohhil, kui tulemused ei olnud valideerimisandmetel paranenud viimase 10 epohhi jooksul. Ploki suurus oli 64 ning sarnaselt eelmainitud artiklile kasutati treenimisel helifailide esimest 6 sekundit, mis Ragano jt [36] sõnul on piisav helifaili loomulikkuse analüüsimiseks.

Teises osas kasutati eelnevalt treenitud kaale ja külmutati kõik wav2vec 2.0 kaalud ning treeniti juurde lineaarne närvivõrgu kiht ennustama MOS väärtust. Iga epohhi jooksul hinnati valideerimisandmetel MSE väärtust ja salvestati kaalud, mille puhul MSE oli kõige madalam. Lisaks arvutati hilisema analüüsi läbiviimiseks lõpliku mudeli korrelatsiooni koefitsiendid LCC, SRCC ja KTAU. Mudeli üldistuvust teistele soome-ugri keeltele testiti võrukeelsetel andmetel.

3.3 UTMOSv2 peenhäälestamine

Antud eksperimendi raames kasutati eeltreenitud UTMOSv2 mudelit, mis peenhäälestati eestikeelsetel andmetel. Mudeli kaalud imporditi Hugging Face keskkonnast[13] ja arhitektuur

[12] https://github.com/alessandroragano/scoreq
[13] https://huggingface.co/sarulab-speech/UTMOSv2/tree/main

laaditi alla GitHub'i repositooriumina[14]. UTMOSv2 treenimisel kasutati ristvalideerimist, mistõttu on avalikult kättesaadavad viis erinevat kaalufaili. Käesolevas töös kasutati mudeli peenhäälestamiseks kaalufaili nimega *fold0_s42_best_model.pth*.

Mudeli peenhäälestamisel kasutati kaofunktsiooni, mida on kirjeldatud töö teoreetilises osas valemis (3). Funktsiooni parameetrite λ_{con}, λ_{mse} ja α väärtused jäeti samaks, mida kasutati treeningprotsessis. Rakendati optimeerijat AdamW õpisammuga 1×10^{-4} ja õpisammu ajastajat CosineAnnealingLR [30]. Mudeli peenhäälestamisel kasutati audiofailidest esimest 10 sekundit. Peenhäälestamisel oli ploki suurus 4 ja peenhäälestamine lõpetati 33. epohhil, kui MSE ei olnud valideerimisandmetel paranenud viimase 10 epohhi jooksul. Salvestati kaalud, kus valideerimisandmete MSE oli kõige madalam. Hilisemaks analüüsiks arvutati valideerimisandmetel LCC, SRCC ja KTAU.

Mudeli üldistuvust teistele soome-ugri keeltele testiti võrukeelsetel andmetel. UTMOSv2 arhitektuurist tulenevalt ei loo mudel testimisel seni tundmatule andmestikule uut unikaalset tunnusvektorit. Artikli autorite soovitusel [30] kasutati mudeli testimisel treeningprotsessis esinenud kolme andmestiku tunnusvektoreid, ennustati ühe helifaili kohta kolm MOS väärtust ja arvutati keskmine.

[14] https://github.com/sarulab-speech/UTMOSv2/tree/main

4. Tulemused

Selles peatükis antakse ülevaade töö praktilises osas loodud sünteeskõne hindamismudelite tulemustest nii eesti- kui ka võrukeelsetel andmetel. Esitatakse 95% usaldusvahemikud võrukeelsete sünteeskõne mudelite hindamistulemustest ja analüüsitakse hindamismudelite üldistusvõimet teistele soome-ugri keeltele. Kõigi võrukeelsete mudelite MOS väärtused on leitud 100 hinnatud audiofaili põhjal.

4.1 Tulemused eestikeelsetel valideerimisandmetel

Tabel 2. Tulemused eestikeelsetel valideerimisandmetel

Mudel	MSE ↓	LCC ↑	SRCC ↑	KTAU ↑
wav2vec 2.0	0.352	0.669	0.630	0.464
SCOREQ	**0.224**	**0.802**	0.768	0.589
UTMOSv2	0.230	0.797	**0.784**	**0.600**

Tabelisse 2 on koondatud mudelite keskmise ruutvea (MSE) ja kolme korrelatsioonikordaja (LCC, SRCC, KTAU) tulemused eestikeelsetel valideerimisandmetel. Parimad tulemused hindamismeetodite kaupa on välja toodud paksemas kirjas. SCOREQ kaofunktsioonil treenitud mudel (edaspidi SCOREQ) saavutas madalaima MSE ja kõrgeima LCC, sealjuures erinevused UTMOSv2 mudeliga jäävad alla 0.01. Seega võib kahte mudelit lugeda antud näitajate alusel võrdväärseks. Samas UTMOSv2 mudeli korrelatsioonikordajad SRCC ja KTAU on saavutanud parema tulemuse, mis ei ole ootuspärane, sest SCOREQ mudel on treenitud kasutades korrelatsioonipõhist kaofunktsiooni. Kuna käesoleva töö eesmärk on luua mudel, mis on võimeline järjestama erinevaid TTS mudeleid MOS väärtuse alusel, on oluline mudeli suutlikus eristada MOS väärtuste suhtelist järjestust. Mudeli wav2vec 2.0 hindamistulemused jäävad oluliselt alla SCOREQ ja UTMOSv2 mudelitele. Seega eestikeelse sünteeskõne hindamisel annavad arvestatava tulemuse suurema tõenäosusega SCOREQ kaofunktsioonil treenitud mudelid wav2vec 2.0 ja UTMOSv2.

4.2 Tulemused võrukeelsetel testandmetel

Tabel 3. Tulemused võrukeelsetel testandmetel

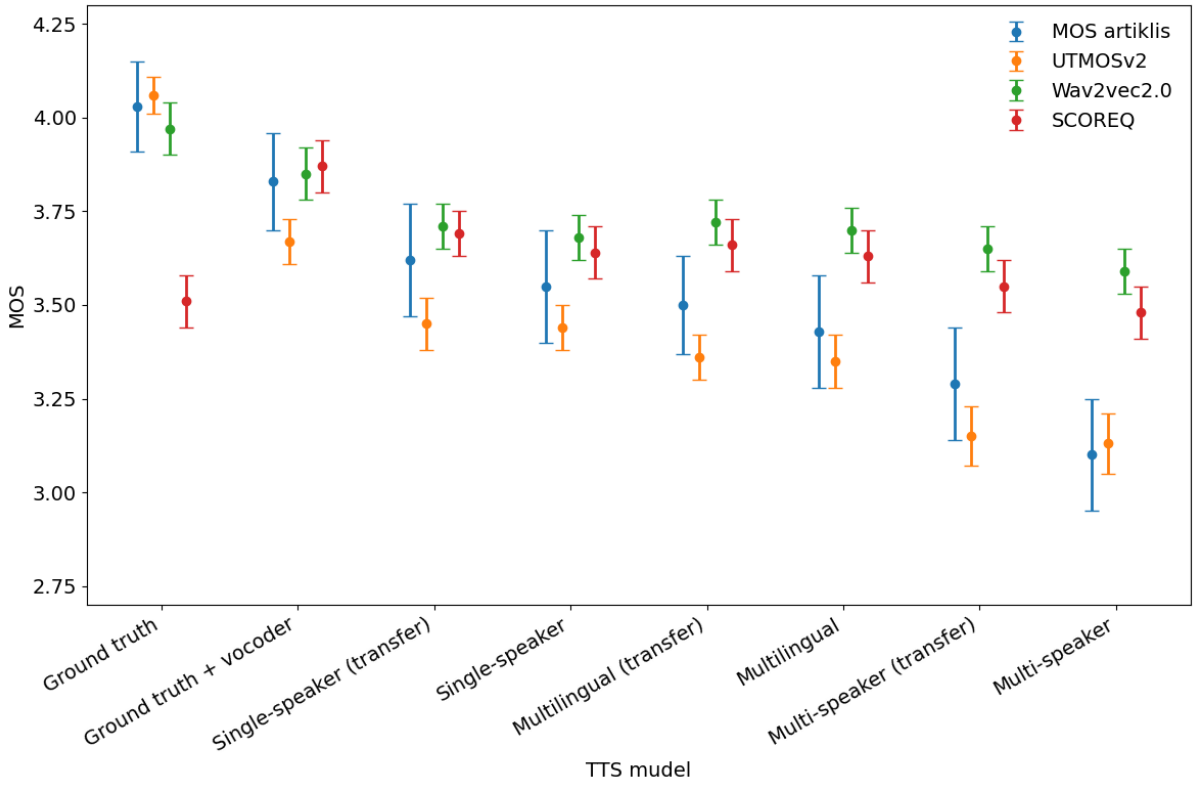
Mudel	MSE ↓	LCC ↑	SRCC ↑	KTAU ↑
wav2vec 2.0	0.639	0.177	0.168	0.118
SCOREQ	0.632	0.202	0.191	0.132
UTMOSv2	**0.606**	**0.349**	**0.311**	**0.220**

Tabelisse 3 on koondatud mudelite keskmise ruutvea ja kolme korrelatsioonikordaja tulemused võrukeelsetel testandmetel. Parimad tulemused hindamismeetodite kaupa on toodud paksemas kirjas. Tulemused võrukeelsetel andmetel on kõigi mudelite puhul tunduvalt madalamad kui eestikeelsetel andmetel. Sellegipoolest eristub selgelt mudel UTMOSv2 just korrelatsioonikordajate poolest. Seetõttu üldistub tundamtutele soome-ugri keeltele UTMOSv2 kõige paremini.

Tabel 4. Võrukeelsete mudelite keskmise arvamuse skoor koos 95% usaldusvahemikuga

Mudel	MOS	wav2vec2.0	SCOREQ	UTMOSv2
Ground truth	4.03 ± 0.12	3.97 ± 0.07	3.51 ± 0.07	4.06 ± 0.05
Ground truth + vocoder	3.83 ± 0.13	3.85 ± 0.07	3.87 ± 0.07	3.67 ± 0.06
Single-speaker (transfer)	3.62 ± 0.15	3.71 ± 0.06	3.69 ± 0.06	3.45 ± 0.07
Single-speaker	3.55 ± 0.15	3.68 ± 0.06	3.64 ± 0.07	3.44 ± 0.06
Multilingual (transfer)	3.50 ± 0.13	3.72 ± 0.06	3.66 ± 0.07	3.36 ± 0.06
Multilingual	3.43 ± 0.15	3.70 ± 0.06	3.63 ± 0.07	3.35 ± 0.07
Multi-speaker (transfer)	3.29 ± 0.15	3.65 ± 0.06	3.55 ± 0.07	3.15 ± 0.08
Multi-speaker	3.10 ± 0.15	3.59 ± 0.06	3.48 ± 0.07	3.13 ± 0.08

Tabelis 4 on välja toodud võrukeelsete TTS mudelite võrdlus MOS väärtuse alusel. Mudelite inimhinnangutel põhinevad MOS väärtused koos usaldusvahemikega on võetud Rätsepa jt [47] artiklist. Mudelite detailsed kirjeldused on toodud eelmainitud artiklis. Tabelist 4 on näha, et kõik hindamismudelid on sünteeskõne mudelite järjekorra mõne erinevusega õigesti ennustanud. SCOREQ mudel hindas inimkõne (*Ground truth*) MOS väärtuse üheks kõige madalamaks. Viga võib tuleneda asjaolust, et mudeli peenhäälestamisel kasutati eeltreenitud mudelit, mis oli treenitud ainult sünteeskõne andmestikel. Nii wav2vec 2.0 kui ka SCOREQ hindamismudelid järjestasid *Single-speaker* sünteeskõne mudeli järjestuses madamale. UTMOSv2 järjestas mudelid õigesti.



Joonis 9. 95% usaldusvahemik võrukeelsetel andmetel mudeli tasandil

Joonisel 9 on visualiseeritud tabelis 4 välja toodud MOS väärtuse usaldusvahemikud kõigi hinnatud võrukeelsete TTS mudelite kohta. Jooniselt on näha, et ainsana kattuvad tegelikkusega vaid mudeli UTMOSv2 usaldusvahemikud kõigil võrokeelsetel mudelitel.

Kuna hindamismudelite üldistusvõime testimine viidi läbi vaid võrukeelsetel andmetel ja võru keel on eesti keelele sarnasem kui paljud teised soome-ugri keeled, siis ei saa kindlalt väita, et mudel UTMOSv2 üldistub ka teistele, eesti keelest kaugematele soome-ugri keeltele.

5. Kokkuvõte

Käesolevas töös treeniti ja peenhäälestati kolm sünteeskõne hindamismudelit, milleks olid wav2vec 2.0, SCOREQ kaofunktsioonil eeltreenitud wav2vec 2.0 ning UTMOSv2. Eesmärk oli luua eestikeelset sünteeskõne hindav mudel, mis korreleerub inimese hinnangutega, ning uurida mudeli üldistusvõimet erinevatele soome-ugri keeltele. Mudelite arendamiseks koondati 3 erineva aasta projektide eestikeelsed sünteeskõne hindamistulemused ning mudelite üldistusvõimet testiti võrukeelsetel andmetel.

Testimine näitas, et parima korrelatsiooni ja üldistusvõime saavutas mudel UTMOSv2. UTMOSv2 korrelatsiooninäitajad LCC, SRCC ja KTAU olid teistest mudelitest võrukeelsetel andmetel tunduvalt paremad. Lisaks järjestas UTMOSv2 ainsana kõik testandmestikus esinenud sünteeskõne mudelid MOS väärtuse alusel õigesti. UTMOSv2 oli ka ainuke mudel, mille 95% usaldusvahemikud võrukeelsetel andmetel mudelite kaupa kattusid inimhinnangutel põhinevate usaldusvahemikega.

Kuna mudeli üldistusvõimet soome-ugri keeltele testiti vaid võru keele peal, mis on eesti keelele sarnane keel, ei saa kindlalt väita, et UTMOSv2 üldistub ka eesti keelest kaugematele soome-ugri keeltele. Tulevikus parema üldistusvõime tagamiseks tuleks testida mudelit ka teistel keeltel ja vajadusel täiendada treeningandmestikku ning sooritada uus peenhäälestamine.

Viited

[1] Tan X., Qin T., Soong F. ja Liu T.-Y. A Survey on Neural Speech Synthesis. 2021. https://arxiv.org/abs/2106.15561.

[2] Wester M., Valentini-Botinhao C. ja Henter G. Are we using enough listeners? No! An empirically-supported critique of Interspeech 2014 TTS evaluations. 2015. DOI: 10.21437/Interspeech.2015-689.

[3] Goodfellow I., Bengio Y. ja Courville A. Deep Learning. MIT Press, 2016.

[4] Cornell University. Neural Networks and Machine Learning. 2015. https://blogs.cornell.edu/info2040/2015/09/08/neural-networks-and-machine-learning/ (09.04.2025).

[5] GeeksforGeeks. What is a Neural Network? 2025. https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/ (09.04.2025).

[6] Cota S. Deep Learning Basics — Part 7 — Feed Forward Neural Networks (FFNN). 2023. https://medium.com/@sasirekharameshkumar/deep-learning-basics-part-10-feed-forward-neural-networks-ffnn-93a708f84a31 (08.12.2024).

[7] Yathish V. Loss Functions and Their Use In Neural Networks. 2022. https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9/ (01.05.2025).

[8] Tan X. Neural Text-to-Speech Synthesis. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer Singapore, 2023. DOI: 10.1007/978-981-99-0827-1.

[9] Sügis E., Tampuu A., Aljanaki A., Fišel M. ja Kull M. Praktiline andmeteadus. Kõrgkooliõpik. Tartu Ülikooli arvutiteaduse instituut, 2024. https://courses.cs.ut.ee/b/andmeteadus.

[10] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L. ja Polosukhin I. Attention Is All You Need. 2023. arXiv: 1706.03762. https://arxiv.org/abs/1706.03762.

[11] Allamar J. Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention). 2018. https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/ (08.12.2024).

[12] Allamar J. The Illustrated Transformer. 2018. https://jalammar.github.io/illustrated-transformer/ (08.12.2024).

[13] Li S. Understanding Positional Encoding in Transformers and Beyond with Code. 2024. https://medium.com/%40lixue421/understanding-positional-encoding-in-transformers-2c7336728be5 (11.02.2025).

[14] Hugging Face. BERT. https://huggingface.co/docs/transformers/en/model_doc/bert (11.04.2024).

[15] Abaskohi A. Navigating Transformers: A Comprehensive Exploration of Encoder-Only and Decoder-Only Models, Right Shift, and Beyond. 2023. https://medium.com/@amirhossein.abaskohi/navigating-transformers-a-comprehensive-exploration-of-encoder-only-and-decoder-only-models-right-a0b46bdf6abe (11.04.2024).

[16] Boigne J. An Illustrated Tour of Wav2vec 2.0. 2023. https://jonathanbgn.com/2021/09/30/illustrated-wav2vec-2.htm (21.04.2025).

[17] Lakdari M. W., Ahmad A. H., Sethi S., Bohn G. A. ja Clink D. J. Mel-frequency cepstral coefficients outperform embeddings from pre-trained convolutional neural networks under noisy conditions for discrimination tasks of individual gibbons. 2024. DOI: 10.1016/j.ecoinf.2023.102457. https://doi.org/10.1016/j.ecoinf.2023.102457.

[18] Roberts L. Understanding the Mel Spectrogram. 2024. https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53 (23.04.2025).

[19] Network P. N. S. What is a Spectrogram? https://pnsn.org/spectrograms/what-is-a-spectrogram (23.04.2025).

[20] Wolf-Monheim F. Spectral and Rhythm Features for Audio Classification with Deep Convolutional Neural Networks. 2024. arXiv: 2410.06927 [cs.SD]. https://arxiv.org/abs/2410.06927.

[21] Muda L., Begam M. ja Elamvazuthi I. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. 2010. arXiv: 1003.4083 [cs.MM]. https://arxiv.org/abs/1003.4083.

[22] Hunt A. J. ja Black A. W. Unit selection in a concatenative speech synthesis system using a large speech database. *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings* 1 (1996), 373–376 vol. 1. https://api.semanticscholar.org/CorpusID:14621185.

[23] Black A. W., Zen H. ja Tokuda K. Statistical parametric speech synthesis. IEEE, 2007.

[24] Viswanathan M. ja Viswanathan M. Measuring speech quality for text-to-speech systems: development and assessment of a modified mean opinion score (MOS) scale. *Computer Speech & Language* 19.1 (2005), lk 55–83.

[25] Do P., Coler M., Dijkstra J. ja Klabbers E. Resource-Efficient Fine-Tuning Strategies for Automatic MOS Prediction in Text-to-Speech for Low-Resource Languages. 2023. arXiv: 2305.19396 [eess.AS]. https://arxiv.org/abs/2305.19396.

[26] Lo C.-C., Fu S.-W., Huang W.-C., Wang X., Yamagishi J., Tsao Y. ja Wang H.-M. MOSNet: Deep Learning-Based Objective Assessment for Voice Conversion. *Interspeech 2019*. ISCA, 2019. DOI: 10.21437/interspeech.2019-2003.

[27] Cooper E., Huang W.-C., Toda T. ja Yamagishi J. Generalization Ability of MOS Prediction Networks. 2022. arXiv: 2110.02635. https://arxiv.org/abs/2110.02635.

[28] VoiceMOS Challenge 2024. https://sites.google.com/view/voicemos-challenge/past-challenges/voicemos-challenge-2024 (23.04.2025).

[29] Huang W.-C., Cooper E., Tsao Y., Wang H.-M., Toda T. ja Yamagishi J. The VoiceMOS Challenge 2022. 2022. arXiv: 2203.11389 [cs.SD]. https://arxiv.org/abs/2203.11389.

[30] Baba K., Nakata W., Saito Y. ja Saruwatari H. The T05 System for The VoiceMOS Challenge 2024: Transfer Learning from Deep Image Classifier to Naturalness MOS Prediction of High-Quality Synthetic Speech. 2024. arXiv: 2409.09305 [cs.SD]. https://arxiv.org/abs/2409.09305.

[31] Baevski A., Zhou H., Mohamed A. ja Auli M. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. 2020. arXiv: 2006.11477 [cs.CL]. https://arxiv.org/abs/2006.11477.

[32] Tsang S.-H. Brief Review — wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. 2014. https://sh-tsang.medium.com/brief-review-wav2vec-2-0-a-framework-for-self-supervised-learning-of-speech-representations-9b9a8fdab85e#:~:text=,supervised%20objective (01.05.2025).

[33] Platen P. von. Fine-Tune Wav2Vec2 for English ASR with Transformers. 2021. https://huggingface.co/blog/fine-tune-wav2vec2-english?utm_source=chatgpt.com (14.04.2024).

[34] Chen L.-W. ja Rudnicky A. Exploring Wav2vec 2.0 fine-tuning for improved speech emotion recognition. 2023. arXiv: 2110.06309 [eess.AS]. https://arxiv.org/abs/2110.06309.

[35] Cooper E., Huang W.-C., Tsao Y., Wang H.-M., Toda T. ja Yamagishi J. The VoiceMOS Challenge 2023: Zero-shot Subjective Speech Quality Prediction for Multiple Domains. 2023. arXiv: 2310.02640 [eess.AS]. https://arxiv.org/abs/2310.02640.

[36] Ragano A., Skoglund J. ja Hines A. SCOREQ: Speech Quality Assessment with Contrastive Regression. 2025. arXiv: 2410.06675 [cs.SD]. https://arxiv.org/abs/2410.06675.

[37] Tan M. ja Le Q. V. EfficientNetV2: Smaller Models and Faster Training. 2021. arXiv: 2104.00298 [cs.CV]. https://arxiv.org/abs/2104.00298.

[38] Kent State University Libraries. SPSS tutorials: Pearson correlation. 2025. https://libguides.library.kent.edu/SPSS/PearsonCorr (10.04.2025).

[39] Swapnilbobe. Spearman's Correlation. 2021. https://medium.com/analytics-vidhya/spearmans-correlation-f34c094d99d8 (10.04.2024).

[40] Beyond D. S. Understanding Kendall's Tau Rank Correlation. 2003. https://ishanjainoffical.medium.com/understanding-kendalls-tau-rank-correlation-c959a7daea56 (10.04.2024).

[41] The Editors of Encyclopaedia Britannica. Finno-Ugric languages. Encyclopaedia Britannica, 20. september 2022. https://www.britannica.com/topic/Finno-Ugric-languages (10.04.2025).

[42] Eesti entsüklopeedia. Soome-ugri keeled. 2006. https://entsyklopeedia.ee/entry/soome-ugri_keeled (10.04.2025).

[43] Võru Instituut. Võru keel. https://wi.ee/et/voru-keel/ (10.05.2024).

[44] Rätsep L., Piits L., Pajupuu H., Hein I. ja Fišel M. Neural Speech Synthesis for Estonian. 2020. arXiv: 2010.02636 [cs.CL]. https://arxiv.org/abs/2010.02636.

[45] Rätsep L., Lellep R. ja Fišel M. Estonian Text-to-Speech Synthesis with Non-autoregressive Transformers. 2022. https://api.semanticscholar.org/CorpusID:252460909.

[46] Rätsep L., Lellep R. ja Fishel M. Enabling Conversational Speech Synthesis using Noisy Spontaneous Data. *Proc. Interspeech 2024*. 2024, lk 4923–4927.

[47] Rätsep L. ja Fišel M. Neural Text-to-Speech Synthesis for Võro. 2023. https://aclanthology.org/2023.nodalida-1.73/.

[48] Ping W., Peng K., Gibiansky A., Arik S. O., Kannan A., Narang S., Raiman J. ja Miller J. Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning. 2018. arXiv: 1710.07654 [cs.SD]. https://arxiv.org/abs/1710.07654.

[49] Łańcucki A. FastPitch: Parallel Text-to-speech with Pitch Prediction. 2021. arXiv: 2006.06873 [eess.AS]. https://arxiv.org/abs/2006.06873.

[50] GeeksforGeeks. Difference Between Autoregressive And Non-Autoregressive Models. https://www.geeksforgeeks.org/difference-between-autoregressive-and-non-autoregressive-models/ (23.04.2025).

[51] University of Tartu. UT Rocket. 2018. DOI: 10.23673/PH6N-0144.

[52] Yadav H. Dropout in Neural Networks. 2022. https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9/ (10.05.2024).

Lisad

5.1 Peenhäälestatud mudeli UTMOSv2 kaalud

Peenhäälestatud mudeli UTMOSv2 kaalud on kättesaadavad Hugging Face repositooriumis leheküljel https://huggingface.co/monatolmats/utmosv2-estonian/tree/main.

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Mona Tolmats,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „Automaatne kõnesünteesi kvaliteedi hindamine soome-ugri keeltele", mille juhendaja on Liisa Rätsep, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;
2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commonsi litsentsiga CC BY NC ND 4.0, 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.

Mona Tolmats

15.05.2025