TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Tiia Sildam 221526IAPM
Andra Velve 221494IAPM

KASKAAD- JA OTSEMEETODI VÕRDLUS EESTI KEELE SUULISE KÕNE TÕLKESÜSTEEMIDE NÄITEL

Magistritöö

Juhendaja: Tanel Alumäe
PhD

Tallinn 2024

Autorideklaratsioon

Kinnitame, et oleme koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autorid: Tiia Sildam, Andra Velve

04.06.2024

Annotatsioon

Kaskaad- ja otsemeetodi võrdlus eesti keele suulise kõne tõlkesüsteemide näitel

Töö eesmärk on teada saada, kui palju erineb kaskaad- ja otsemeetod-mudelite jõudlus erinevatel tõlkesuundadel. Lisaks uuriti peenhäälestamise mõju otsemeetod-mudelitele. Mudelite võrdlused teostati neljal suunal: eesti-inglise, eesti-vene, inglise-eesti ja vene-eesti. Tõlkemootorid, mida kaskaadmudelites masintõlget teostavate osadena käsitleti, olid järgnevad: GPT3.5-turbo, GPT3.5-turbo-instruct, GPT4, Neurotõlge, Google Translate, NLLB 3.3B. Otsemeetod-arhitektuuriga mudelid olid järgnevad: Whisper large v3, SeamlessM4T v2 large ja OWSM 3.1 EBF. Valideerimisandmestikud koosnesid peamiselt vestlussaadetest, pressikonverentsidest ja uudissaadetest. Valideerimisandmed transkribeeriti käsitsi ja transkriptsioonid tõlgiti tõlkebüroode poolt.

Töö tulemusel valmisid otsemeetod-mudelid, mis on võimelised tõlkima käsitletud suundi sarnaselt kaskaadmudelitele. Töö käigus valmis andmestik, mida on võimalik edaspidi kasutada eestikeelse kõne ja teksti tõlkimise süsteemide parendamiseks. Andmestik koosneb sünteetilistest andmetest ja internetist kogutud lisaandmetest. Töö käigus läbiviidud eksperimentidest nähtus, et sünteetiliste andmete kasutamine otsemeetod-mudelite peenhäälestamiseks parandab mudelite jõudlust märgatavalt. Kaskaad- ja otsemeetod-mudelite jõudlused erinesid mõningal määral olenevalt tõlkesuunast, kuid statilist olulisust üldiselt ei esinenud.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 72 leheküljel, 11 peatükki, 23 joonist, 11 tabelit.

Abstract

Comparing Cascaded and End-to-End Approaches for Estonian Spoken Language Translation

The aim of the study was to compare cascaded and direct models in the context of Estonian spoken language translation. To achieve this, existing speech translation datasets were analyzed, and an overview of the representation of the Estonian language in these datasets was created. During the study, data was synthesized by translating text files made by automatic speech recognition into the target language, and additional data was collected from the web. Subsequently, experiments were conducted with various cascaded and direct models, with direct models evaluated in both out of the box and finetuned versions. Results were evaluated using automatic metrics BLEU and BLEURT. Study focused on bidirectional Estonian-English and Estonian-Russian conversational speech translation.

Cascaded models consisted of automatic speech transcription (ASR) system paired with translation models. Cascaded models used two ASR models: Estonian-X direction used Whisper-medium-et-orthographic, X-Estonian direction used Whisper-large-v3. The chosen translation engines were: GPT3.5-turbo, GPT3.5-turbo-instruct, GPT4, Neurotõlge, Google Translate, NLLB-200 3.3B. The direct models were: Whisper large v3, SeamlessM4T v2 large, and OWSM 3.1 EBF. Model comparisons were conducted in all chosen translation directions. Validation datasets consisted mainly of talk shows, press conferences and news broadcasts. Validation data was transcribed manually, and the transcripts were translated by translation agencies.

The study found that models performance varied slightly by translation direction, with Estonian-X achieving higher scores than X-Estonian. Both cascaded and fine-tuned end-to-end models had similar validation results, with minor differences in BLEU scores. BLEURT metrics also showed small performance differences. A Wilcoxon signed-rank test was conducted on validation data for three models: Whisper + Google Translate, finetuned Whisper-large-v3, and finetuned SeamlessM4T large. It indicated that the fine-tuned SeamlessM4T model had the best overall performance, since there was no statistical significance for other translation directions and it outperformed both the cascaded system

and finetuned Whisper in the Estonian-Russian direction. Fine-tuning direct models solely on synthetic data had a notably positive effect on validation scores. However, when scraped web data was combined with synthetic data, the results often did not improve and, in some cases, even worsened.

As a result of the study, a dataset was created that can be used to improve Estonian speech and text translation systems in the future. The dataset consists of synthetic data and additional data collected from the Internet. Synthetic data includes approximately 1300 hours in the Estonian-English and Estonian-Russian direction. In other directions it contains about 1000 hours for English-Estonian translation and 102 hours for Russian-Estonian translation. Web data comprises 40 hours in the Estonian-English direction, 18 hours in the Estonian-Russian direction, 58 hours in the English-Estonian direction and 617 hours in the Russian-Estonian direction. Experiments conducted during the study showed that the use of synthetic data significantly improves the performance of the models.

The thesis is written in Estonian and is 72 pages long, including 11 chapters, 23 figures and 11 tables.

Lühendite ja mõistete loetelu

API	*Application Programming Interface*, rakendusliides ehk programmiliides
ASR	*Automatic speech recognition*, automaatne kõnetuvastus
BART	*Bidirectional and Auto-Regressive Transformers*
BERT	*Bidirectional Encoder Representations from Transformers*
BLEU	*Bilingual Evaluation Understudy*
Dekooder	Süvaõppemudeli osa, mis genereerib vektorist vastava väljundi
E2E	*End to end*, otsast lõpuni ehk otsesüsteem
Endpoint	Lõpp-punkt ehk liides, mille abil protsessid saavad informatsiooni vahetada
Feed-forward network	Pärilevivõrk
GPT	*Generative Pre-trained Transformer*
GRU	*Gated recurrent unit*
Kooder	Süvaõppemudeli osa, mis kaardistab muutuva pikkusega lähtejada fikseeritud pikkusega vektoriks
LSTM	*Long Short-Term Memory*, pikk lühiajaline mälu
mBART	*Multilingual BART*
Mudel	Masinõppe algoritm, mis on võimeline leidma ja õppima andmestikus olevaid mustreid
RNN	*Recurrent Neural Network*, rekurrentne närvivõrk
Seq2Seq	*Sequence-to-Sequence*, masinõppe lähenemine, mille abil saab ühte järjestikust jada teisendada teiseks väljundjadaks
Transformer	Süvaõppemudeli arhitektuur, mis põhineb tähelepanumehhanismidel
WebVTT	*Web Video Text Tracks Format*, ajastatud tekstiridade (näiteks subtiitrite) kuvamise vorming

Sisukord

Joonised

Tabelid

1. Sissejuhatus

Umbes 20% inimestest kannatab mingisuguse kuulmisvaeguse all [1]. Kvaliteetne kõneteksti tõlge, mis hõlmab lähtekeelse kõne tõlkimist tekstikujule teise keelde, on vajalik selleks, et need inimesed saaksid samuti osa meediast ja muust informatsioonist nagu tavalise kuulmisega inimesed. Hea kõnetõlge loob eelduse sünkroon-kõnetõlke süsteemide loomiseks, mis võimaldaksid kuulmisvaegusega inimestel vaadata näiteks reaalajas uudiseid. Kõne-teksti tõlke abil on võimalik vähendada inimeste käsitsi tehtavat tööd koosolekute või istungite protokollimisel, kuna selle käigus viiakse kõne teksti kujule ning seeläbi kaob vajadus eraldi transkribeerimise jaoks.

Kõnetõlke arenedes on võimalik seda üha rohkem igapäevaselt kasutada. Keeruliseks teeb kõnetõlke see, et helisalvestistel on tihti taustamüra, keel on keeruline ning masinõppel on muuhulgas raske aru saada sõnade piiridest [2]. Seetõttu otsitakse aina paremaid mudeleid ja meetodeid, mille abil kõnetõlke kvaliteeti tõsta.

Seni on peamiselt kasutusel olnud *cascaded* ehk kaskaadlähenemine. Selle käigus transkribeeritakse lähtekeelne kõne tekstiks ning seejärel tõlgitakse transkriptsioonid sihtkeelde. Kaskaadlähenemise peamine eelis seisneb võimes ära kasutada iga alamkomponendi edusamme. Kaskaadmudelitel esineb ka erinevaid puuduseid, näiteks kõneteabe (näitena: prosoodia) kadumine või vigade levimine automaatse kõnetuvastuse süsteemilt tõlkemudelile. Eelkirjeldatud probleemide tõttu võivad kaskaadmudelid vahel väljastada kehva kvaliteediga tõlkeid [3, 4].

Otsemeetodil põhinev kõnetõlge võib potentsiaalselt parandada eespool kirjeldatud probleeme, mis kaskaadmudelites esinevad. Samuti on alust arvata, et *end-to-end* ehk otselähenemisel põhinev kõnetõlge võib teatud olukordades toimida sama hästi või isegi paremini kui kaskaadmudelid [3]. Erinevalt kaskaadmeetodist ei hõlma otsemeetodiga tehtav kõnetõlge eraldi samme kõnetuvastuseks ja masintõlkeks. Otsemeetodi puhul tehakse need sammud ilma vaheetappideta ning lähtekeelne kõne tõlgitakse otse sihtkeelde [3].

2. Eesmärk

Loomuliku kõne ja keele töötlus on hetkel kiiresti arenev valdkond. Andmehulki ja mudeleid uuendatakse ja parendatakse pidevalt. Kõnetõlke arenedes on võimalik seda üha rohkem igapäevaselt kasutada. Kõnetõlke katseid on tehtud võrdlemisi palju laialdaselt levinud keeltel, näiteks inglise, prantsuse ja hispaania keel. Tihti on tõlkesuunaks just x keele tõlkimine inglise keelde. Vähese ressursiga keeltel pole niivõrd palju katseid tehtud ning hetkel ei ole levinud ka tõlkesuundade jõudluse mõõtmine kahe keele vahel, millest kumbki pole inglise keel [5].

Antud juhul keskendutakse kaskaad- ja otsast lõpuni arhitektuuriga mudelite võrdlemisele. Mudeleid testitakse neljal tõlkesuunal: eesti-inglise, eesti-vene, inglise-eesti ja vene-eesti. Kõigil tõlkesuundadel on olemas professionaalsete tõlkijate poolt tõlgitud valideerimisandmestikud ning mudeleid võrreldakse BLEU ja BLEURT skooridega. Saades teada, et milline mudeliarhitektuur on eesti keele tõlkimisel kõige võimekam, on edaspidi võimalik märgatavalt parandada eestikeelse kõne tõlkemudeleid.

Seda uurimistööd motiveerisid mitmed asjaolud. Esiteks on eesti keele kontekstis treeningandmeid väga vähe. Andmekorpused, mis eesti keelt sisaldavad, sisaldavad tihti seda vaid loetud tundide hulgas ning valdavatel juhtudel ühesuunaliselt (näiteks ainult inglise-eesti suunal). Nendest vähestest korpustest, kus on eesti keel, on enamik dikteeritud - sellised andmed loomuliku kõne tõlkimiseks ei sobi, sest loomulik kõne erineb dikteeritud kõnest märgatavalt. Teiseks on seni tehtud uuringud, mis võrdlevad kaskaad- ja otsemeetodil kõne tekstiks tõlkimist, keskendunud peamiselt laialdase kättesaadavusega keeltele. Eesti keelel on mitmeid eripärasid, mis teeb selle kõnetõlke keeruliseks.

2.1 Uurimisküsimused

Töö eesmärk on teada saada, kui palju erineb kaskaad- ja otsemudelite jõudlus eesti keele tõlkimisel inglise ja vene keelde ning vastupidi. Peenhäälestatud otsemudeleid võrreldakse baasmudelitega selleks, et leida, kui palju on võimalik avalike otsast lõpuni mudelite tulemusi peenhäälestamisega parendada. Käesoleva töö kontekstis on baasmudelid mudelid, mis olid juba eelnevalt olemas ja kasutajatele saadaval ehk erinevate transkribeerimissüsteemide kombineerimine tõlkemootoritega ja peenhäälestamata Whisper, SeamlessM4T ja OWSM mudelid.

Eesti keele kohta pole varem sellisel kujul süsteemset uurimist tehtud. Eesti keel erineb häälikute, häälduse, morfoloogia ja muu sellise poolest teistest valdkonnas peamiselt käsitletud keeltest.

Antud töö uurimisküsimused on järgnevad:

1. Kui hästi töötavad olemasolevad avalikud eeltreenitud kõnetõlkemudelid eesti keele puhul?
2. Kas avalikult saadaolevate andmetega (näiteks subtitreeritud videod internetis) on võimalik avalike eeltreenitud mudelite eesti keele kvaliteeti parandada?
3. Kas sünteesitud andmetega (masintõlgitud kõnetuvastuse treeningandmed) on võimalik avalike eeltreenitud mudelite eesti keele kvaliteeti parandada?
4. Kui hästi toimivad saadud otsast-lõpuni mudelid võrrelduna kaskaadsüsteemiga?

2.2 Kõnetõlke keerukused

Kõnetõlkel on nüansid, millega tuleb arvestada. Lähtekeelse helisignaali kaardistamine sihtkeelseks tekstiks on keeruline, sest kõne võib varieeruda suurel määral, helisalvestistel võib esineda taustamüra ja sõnade piiridest on tihti raske aru saada.

2.2.1 Keelte omadused

Keelte omadused varieeruvad olenevalt keelepaarist palju. Mõndades keeltes on sõnade järjekord lauses väga kindlalt paigas, seevastu teistes keeltes saab ühte lauset samade sõnadega konstrueerida mitmel viisil jättes lause sisulise tähenduse samaks. Keeltel on erinevused häälduses ja ortograafias - näiteks eesti keelt kirjutatakse sarnaselt selle häädusele, kuid inglise keele puhul erineb häädus ja õigekiri oluliselt. Eesti-inglise keelepaarile on iseloomulik ka see, et kui inglise keeles on ruumiliste või ajaliste suhete väljendamiseks prepositsioonid (*at, on, in* jms), siis eesti keeles väljendatakse neid suhteid enamasti käänete abil. Vene ja eesti keelepaari puhul on olemas näiteks ajavormide erinevused - vene keeles eristatakse lõpetatud ja lõpetamata tuleviku tegevusi, kuid eesti keeles pole tuleviku ajavormi eraldi olemas. Eelnevalt on toodud vaid mõned konkreetsed näited keelte vahelistest erinevustest grammatika ja struktuuri poolest. Selliste sisuliste ja vormistuslike erinevuste ületamine on üks põhjustest, miks tõlkimine ja kõnetõlge konkreetsemalt on keeruline.

2.2.2 Tehnilised detailid

Kõne tõlkimisel on tehnilised aspektid, mis vajavad eraldi tähelepanu. Pideval akustilisel signaalil on märkimisväärne varieeruvus ja muutuvus. Suure varieeruvuse tõttu on vaja abstraktsiooni mitmetest elementidest, sealhulgas kõneleja omadustest, murretest, taustamürast ning näiteks ka kanaliomadustest. Kõnetõlkele lisab keerukust asjaolu, et lausete teise keelde tõlkimisel tekib sageli vajadus eraldada asesõnu, kohandada sõnade ja fraaside järjestust, täpsustada sõnade täpset tähendust. Mõnes keeles on mitu korda suurem sõnavara kui teises, sellistes olukordades on oluline leida parim võimalik vaste ja lausekonstruktsioon, et sisu jääks lähtelausega põhimõtteliselt samaks. Lisaks tuleb luua semantiliselt ja süntaktiliselt sobivaid väljundeid [2].

2.2.3 Semantiline korrektsus

Kogu kõnetõlke protsessi käigus on oluline pöörata tähelepanu väljundi korrektsusele ja asjakohasusele, et tõlgitud tekst annaks edasi algset mõtet täpselt ja selgelt. Kõnekeele eripärad ilmnevad sageli häiretes, mõttepausides, vigades, juhuslikus stiilis ja kaudses suhtluses. Kõnekeele ja spontaanse või poolspontaanse kõne tõlkimine on keerulisem ülesanne kui dikteeritud või loetud kõne tõlkimine. Suuline kõne eristub kirjakeelest, millele on omane selgesõnalisus, formaalsus ning grammatiline korrektsus. Kui kõnekeelt tõlgitakse kirjalikku vormi, toimub keeruline üleminek suulisest suhtlusest kirjaliku suhtluse valdkonda. Tõlkimise käigus seistakse silmitsi väljakutsega ületada nende kahe keelekasutuse vaheline lõhe ning väljundina luua tõlgitud tekst, mis peegeldaks täpselt ja mõtestatult originaalmõtet. Tõlkimisel tuleb erilist tähelepanu pöörata mitmetele keelelistele nüanssidele, tagamaks, et tõlgitud tekst oleks mitte ainult sõnaselge, vaid ka kultuuriliselt ja kontekstiliselt adekvaatne [2].

2.2.4 Andmete puudus

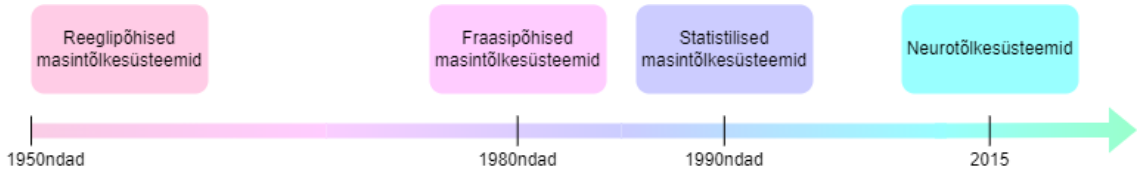
Otsast lõpuni kõnetõlke puhul on väga oluline andmete maht. Sügavad närvivõrgud vajavad suurt hulka andmeid, et neil oleks võimalik õppida tundma ära andmetes sisalduvaid mustreid. Kõnetõlkeandmete märgendamine on nõudlik ja ajakulukas protsess, seetõttu on hetkel treenimiseks sobivaid paralleelandmeid võrdlemisi vähe. Näiteks automaatse kõnetuvastuse (ASR) andmestik, nagu Librispeech [6], sisaldab 960 tundi kõnet ning masintõlke andmestik miljoneid paralleeltekste. Samas kõnetõlke andmestik, nagu MuSTC [7], sisaldab ainult umbes 400 tundi kõnet 230 000 lausungiga. Andmete nappuse tõttu jäävad otsemudelid oma jõudluses kaskaadsüsteemidele alla, sest viimased on treenitud rohkete automaatse kõnetuvastuse ja masintõlke andmetega.

3. Taust

Tõlkimine on vajalik inimestevahelise suhtluse, teadmiste leviku ja kultuuride rikastamise seisukohalt. See aitab ületada keelebarjääre ja edendab ülemaailmset koostööd ning üksteise mõistmist. Käsitsi tõlkimine on aeganõudev, nüüdseks on laialdaselt kasutusel masintõlkesüsteemid, mis aitavad protsessi lihtsustada. Vaatamata sellele on kõnetõlge siiani keeruline.

3.1 Masintõlke areng

Tekstitõlke probleemiga on nüüdseks tegeldud aastakümneid. Masintõlke arengu saab laias laastus jagada neljaks. Arengu etapid on kujutatud Joonisel 1.



Joonis 1. Masintõlke ajalugu.

3.1.1 Reeglipõhine masintõlge

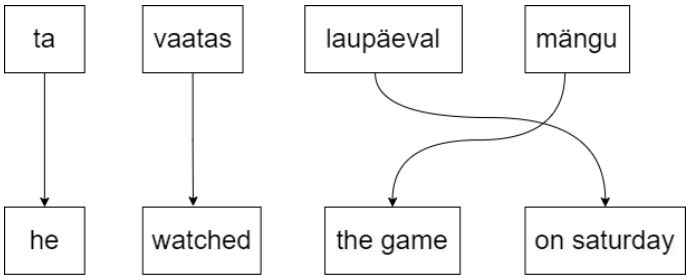
Algselt oli peamine lähenemine reeglipõhiste süsteemide loomine. Kuna keelte grammatika on keeruline ja varieerub erinevate keelte vahel palju, tuli luua suuri ja keerulisi reeglisüsteeme. Reeglisüsteemid koosnesid paljudest ümberkirjutamise reeglitest. Sellegipoolest olid reeglipõhised süsteemid võrdlemisi paindlikud - mudeli arenedes ning uue andmestiku lisandudes oli võimalik reegleid ja mustreid mudelile juurde lisada. Töötlemise tõhususe aspektist olid need siiski ebasoodsad - tõlkimine toimus mustri sobitamise teel otsustuspuu või graafiku peale. Lisaks olid olemas veel teisendamisstruktuurid ja semantilised ülesehitused, mis olid omased konkreetsele keelele. Suur osa töötlemisajast läks mustrite sobitamisele, mille tulemuseks oli tihti võimetus leida mustrile vastav väljund [8].

3.1.2 Fraasipõhine masintõlge

1980ndatel tekkis fraasipõhise masintõlke suund. Selle idee seisnes selles, et kakskeelse andmestiku puhul on võimalik lähtelause ja sihtlause panna omavahel fraaside kaupa nii-öelda sobituma. Nii luuakse fraaside ja nende tõlgete arhiiv, mida kasutatakse teksti tõlkimiseks. Võrreldes reeglipõhiste süsteemidega olid fraasipõhised süsteemid väiksema töötlemise keerukusega ja lihtsama ülesehitusega. Fraasipõhise tõlke puhul oli oluline fraasi pikkuse valimine - kui valiti liiga lühike osa lausest, võis tõlke kvaliteet langeda märgatavalt. Ühe või kahe sõna kaupa tõlkides ei ole süsteemil võimalik lähtekeelse lause sisu säilitada, sest sõna tõlge sõltub oluliselt seda ümbritsevatest sõnadest. Seevastu kui valida liiga pikk fraasi pikkus, ei leia süsteem enam lähtefraasile sobivaid vasteid ning tõlke kvaliteet langeb samuti järsult [9].

3.1.3 Statistiline masintõlge

1990ndatel hakati looma statistilisi masintõlke mudeleid. Tõlge põhines statistilistel mudelitel, millega hinnati lähtekeelse sõna ja sihtkeelse sõna omavahelise sobivuse tõenäosust. Lähtekeelne lause jaotati ühe kuni paari sõna pikkusteks osadeks ning leiti nendele vastavad osad sihtkeelses lauses. Seejärel leiti tõenäosused iga lausepaari sõna-sõnalisele joondumisele. Sõna-sõnalise joondumise näide on kujutatud Joonisel 2. Erinevate lähtekeeles olevate lauseosade tõenäosuseid võrreldes sai leida sihtkeelse väljundlause, mille vastavuse tõenäosus alglausele oli suurim [10].



Joonis 2. Sõna-sõnaline joondumine.

Kui algselt loodi sõnastikud sõna tõlke kaupa, siis sealt edasi arendati süsteeme fraasipõhisemaks. Fraasipõhine lähenemine pakub palju eeliseid, kuna fraasi tõlge haarab oma olemuselt sõna konteksti paremini ja aitab kaasa ka sõnade järjekorra säilitamisele lauses. Lause sõna-sõnalt tõlkimine ei anna eriti häid tulemusi, sest sõna tähendus oleneb oluliselt kontekstist, mis sõna ümbritseb. Lisaks on olemas ülekantud tähendusega fraasid ja ütlused, mille tähendus muutub sõna-sõnalt tõlkides. Niimoodi sõna-sõnalt tõlkides ja sõnastik-

ku luues suureneb sõnastik ka ebavajalike sõnade ja terminite poolest, mis fraasipõhisel tõlkimisel ei pruugi andmestikku lisanduda. Tõlget teostades tuleb sõna-sõnalt tõlkides leida nii palju joondusi kui on sõnu lauses, kuid fraasipõhiselt tuleb leida vaid lähtekeeles olevale fraasile sobiv tõlge. Samuti on lausetes tihti olulise sisuta sõnu, mida muidu ei tõlgitaks. Ka nende sõnadega saab fraasipõhine lähenemine paremini hakkama, jättes need tõlgetest välja [11].

Statistilised masintõlkesüsteemid baseerusid peamiselt ühel või mitmel tõlkemudelil ja sihtkeele keelemudelil. Kuigi oli välja pakutud palju erinevaid tõlkemudeleid ja fraaside eraldamise algoritme, jäi enamikus süsteemides standardiks siiski sõna n-gramm koos taganemise mudeliga. Muutus algas siis, kui hakati pidevaid esitusi kasutama sõnade modelleerimiseks. Schwenk et al. (2006) lõid statistilise keelemudeli, mis põhines sõnade pideva esituse kasutamisel sõnavara loomisel. Seda kasutati närvivõrgu projektsiooni ja tõenäosuse hindamise teostamiseks [12].

3.1.4 Neurotõlge

Umbes 10 aastat tagasi muutus närvivõrkudel baseeruv masintõlge peamiseks suunaks selles valdkonnas. 2013. aastal pakkusid Kalchbrenner ja Blunsom [13] välja uue kooder-dekooder arhitektuuri masintõlke jaoks. Nad tutvustasid tõenäosuslike pidevtõlkemudelite klassi, millele panid nimeks *Recurrent Continuous Translation Models*. Autorid soovisid parandada Schwenk et al. [12] mudelite piiranguid, sest need piirdusid fikseeritud suurusega lähte- ja sihtfraasidega ning lihtsustasid sihtsõnade vahelisi sõltuvusi võttes arvesse piiratud sihtkeele modelleerimisteavet.

Sutskever et al. (2014) pakkusid välja sügavate närvivõrkude kasutamise masintõlkes. Nende meetod kasutas mitmekihilist pikka lühiajalist mälu (LSTM), et kaardistada sisendjada fikseeritud mõõtmetega vektoriga, ja seejärel teist sügavat LSTM-i, et vektorist sihtjada dekodeerida. Nende tulemused näitasid, et neuromasintõlke süsteem, millel on suur sügav LSTM ja piiratud sõnavara, võib ületada standardset statistilise masintõlkepõhist süsteemi. Nad saavutasid oma arhitektuuriga WMT'14 inglise-prantsuse tõlkimise võistlusel parema BLEU skoori kui fraasipõhised statistilise masintõlke süsteemid. See näitas, et suur ja sügav LSTM, millel on piiratud sõnavara ja mis ei tee peaaegu mingeid eelduseid probleemi struktuuri kohta, suudab ületada standardset statistilise masintõlkepõhist süsteemi, mille sõnavara on suuremahulise masintõlke ülesande puhul piiramatu [14].

Eelnevalt kirjeldatud kooder-dekooder arhitektuuriga mudelid olid aluseks uut tüüpi rekurrentsetel närvivõrkudel põhinevatele kooder-dekooder mudelitele. Ühele esimesele sellist laadi mudelile panid aluse Cho et al. 2014. aastal [15]. Samal aastal tegid Bahdanau et al.

[16] kooder-dekooder mudelile laienduse - tähelepanumehhanismi.

Erinevalt traditsioonilistest fraasipõhistest tõlkesüsteemidest, mis koosnevad paljudest väikestest eraldi häälestatud alamkomponentidest, püüab neuraalne masintõlge luua ja treenida ühtset suurt närvivõrku, mis võtab sisendina lause ja väljastab selle tõlke. Enamus väljapakutud närvivõrkudest järgisid kooder-dekooder arhitektuuri, kus iga keele jaoks oli eraldi kooder ja dekooder või kasutati keelespetsiifilist koodrit. Kooder loeb ja kodeerib lähtelause fikseeritud pikkusega vektoriks, seejärel dekooder loeb kodeeritud vektorit ja väljastab vastava tõlke. Koodrist ja dekoodrist koosnev süsteem treenitakse koos, et suurendada õigete tõlgete saamise tõenäosust. Sellel lähenemisviisil on siiski probleem - närvivõrk peab suutma kogu lähtelause vajaliku teabe tihendada fikseeritud pikkusega vektoriks. Kui tegemist on pikkade lausetega, võib närvivõrgu toimetulek olla raskendatud. Eriti keeruline on see, kui sisendiks on laused, mis on treeningkorpuse lausetest pikemad [16].

3.1.5 Hiljutised arengud kõnetõlkes

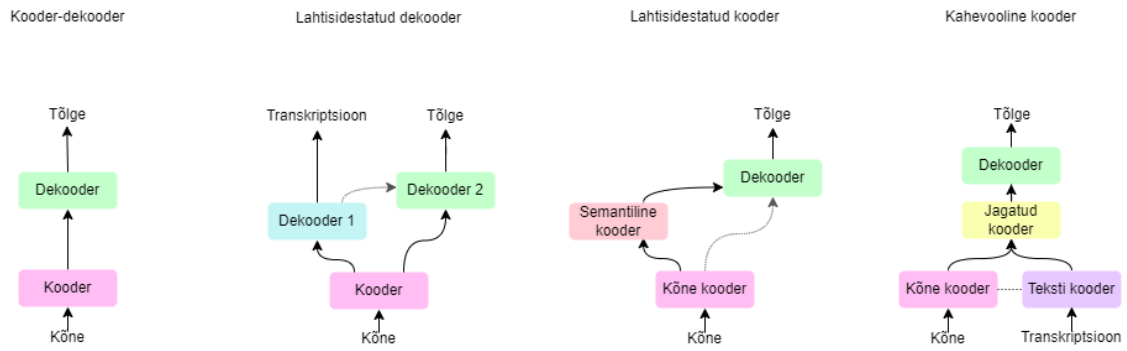
Transformer [17] on hetkel kõige levinum arhitektuur, millega kõnetõlget teostatakse. Sellele on loodud mitmeid edasiarendusi, et veelgi paremaid tulemusi saavutada. Üks selline on näiteks Speech-Transformer [18], mis täiustab transformeri arhitektuuri, lisades arhitektuuri esimeseks osaks sisendi tihendamise tehnika (järjendi helidetailid tihendatakse konvolutsioonikihtide ja normaliseerimiskihiga) ning selle tulemus antakse transformeri koodrile.

Lisaks on olemas kõnetuvastuseks loodud Conformer arhitektuur [19], mis baseerub transformeri arhitektuuril ning ühendab isetähelepanu ja konvolutsioonilise närvivõrgu. Selle peamine omadus on konvolutsioonimoodul, mis pannakse iga koodri mitmepealise isetähelepanu mooduli ja koodriploki pärilevivõrgu kihi vahele. Konvolutsioonimoodul koosneb tähelepanu ja konvolutsiooni osadest, mis on Macaron-net arhitektuuri stiilis kahe pärilevivõrgu ja residuaalsete ühenduste vahel. Konvolutsioonid võtavad paremini arvesse kohalikku teavet, samas kui transformer mudel on hea globaalse teabe ammutamiseks. Selline kombinatsioon aitab paremini kodeerida näiteks pikka kõnet. Conformerist on loodud ka edasiarendus Branchformer [20], mis arendab eelnevat arhitektuuri edasi muutes seda veelgi skaleeritavamaks ja paindlikumaks.

Hetkel on levinud ka transformeri kombineerimine mõne enesejärelevalvega õpitud kõneesituste (*self-supervised speech representations* ehk SSL) raamistikuga, näiteks wav2vec [21, 22] või HuBERT [23]. Kõne omaduste saamiseks antakse algne helilaine antakse SSL mudelisse, mis töötleb heli läbi mitme konvolutsioonilise kihi ja transformeri koodri kihti-

de. SSL mudeli kasutamiseks on mitu varianti: võidakse kasutada SSL mudeleid koodrina ning transformerit dekoodrina või kasutatakse SSL mudelit enne tervet transformerit [24].

Katseid on tehtud ka multitegumraamistikega (*multi-task framework*). Nende idee on teha ülesanne mitmeks osaks, et parandada tulemust. Kõnetõlke puhul on abiülesanneteks sageli ASR ja masintõlge. Mudeli struktuuri osas saavad siht- ja abiülesannete moodulid mõningaid parameetreid jagada, kusjuures moodulite osad ise jäävad üksteisest sõltumatuks. Laias laastus jagunevad multitegumraamistikud kolmeks: lahtisidestatud kooder, lahtisidestatud dekooder ja kahevooline kooder [24].



Joonis 3. Multitegumraamistikud.

Joonisel 3 on kujutatud levinud multitegumraamistike arhitektuurid. Lahtisidestatud dekoodri arhitektuuri idee on leevendada modelleerimiskoormust. Meediumite vaheline ja mitmekeelne modelleerimine kasutab intensiivselt ressursse, et leida andmetes seoseid ja mustreid. Selle leevendamiseks võetakse kasutusele täiendav dekooder, mis juhib transkriptsiooni õppimist. Mudel ise treenitakse endiselt otsast lõpuni stiilis. Lahtisidestatud dekoodri arhitektuur põhjustab siiski keerulisi disaine ja mitme järelduse tegemine toob endaga kaasa kõrge latentsuse ehk ooteaja. Lahtisidestatud kooder võimaldab samaaegselt ära tunda ja mõista algse kõnesisendi semantikat. Kooder jaguneb kaheks - esimene kooder kodeerib esmalt kõnesisendist pärineva akustilise teabe, seejärel õpib semantiline kooder tõlkeks vajamineva semantilise esituse. Lisaks on veel olemas kahevooline kooder, millel on kõne ja teksti kooder. Selle eeliseks on see, et mudel suudab paremaid semantilisi esitusi õppida, kui jagatud kooder saab sisendi nii kõne koodrilt kui ka teksti koodrilt [24].

3.2 Eelnevad tööd

Mitmed uuringud on võrrelnud kaskaad- ja otsast lõpuni süsteemide jõudlust erinevatel keelepaaridel. Peamiselt on küll uuringud keskendunud inglise keelele koos mõne teise

keelega, näiteks otsesüsteemi jõudlust uuriti prantsuse-inglise [25] suunal ja hispaania-inglise [26] suunal.

2021 aastal avaldati artikkel, mille eesmärgiks oli võrrelda kaskaad- ja otsesüsteemi jõudlust. Artikli autorid said inspiratsiooni IWSLT konverentsi 2018 ja 2019 aasta tulemustest - otsast lõpuni süsteemid osalesid konverentsi võistlustel esimest korda 2018ndal aastal. Paari aastaga jõudis kahe lähenemise jõudluse vahe kahaneda mõne BLEU punktini. Artikli autorid võtsid suundadeks inglise-saksa/hispaania/itaalia keeled ning treenisid tolle aja tipptasemel kaskaad- ja otsemudelid. Mudeleid testiti MuST-C korpuse peal. Autorid tõid välja ka seda, et tolle hetkeni oli kahte lähenemist võrreldud tihti tasakaalustamata oludes - näiteks hinnati ainult üht keelepaari ja tugineti ainult automaatsetele mõõdikutele. Antud artiklis võrreldi seetõttu kolme keelepaari ning kasutati lisaks teistele mõõdikutele ka professionaalseid tõlkijaid. Uuringu tulemusel jõuti mitmele järeldusele: kaskaadsüsteemid toimivad antud keelepaaride puhul paremini morfoloogia, sõnajärjekorra ja leksikaalsete mõistete valdkonnas, kuid otsesed süsteemid mõistavad heli paremini ja suudavad prosoodiat paremini edasi anda. Üldiselt jõuti järeldusele, et eelmainitud keelepaaride puhul ei tee enamikel juhtudel inimene enam vahet, kas tõlge on loodud kaskaad- või otsesüsteemi poolt [3].

Ühes 2022 aastal avaldatud artiklis võrreldi kahte lähenemist hispaania ja baski keele vahel. Baski keele puhul on oluline märkida, et tegemist on väheste ressursidega keelega ehk selle keele kohta pole nii palju andmeid kui näiteks hispaania või inglise keele jaoks. Hispaania ja baski keele enda vahel on ka mitmed morfoloogilised ja lausestruktuuri erinevused. Uuringus kasutati mudelite treenimiseks peamiselt mintzai-ST andmekorpust, mis koosneb Baski parlamendi koosolekute salvestustest. Artikli autorid jõudsid järelduseni, et kuigi otsast lõpuni masinõpe on aastate jooksul märgatavalt paranenud, jääb see antud keelepaari puhul siiski kaskaadsüsteemile tulemustes alla [27].

Eelmainitud artiklites on tihti teemaks andmete puudus ning sellest tulenevalt ka otsast lõpuni süsteemide halvem jõudlus. Mitu aastat hiljem on see probleem vähese ressursiga keelte puhul endiselt aktuaalne, kuigi viimastel aastatel on tehtud mitmeid mitmekeelseid andmekorpuseid (NLLB, MuST-C jms). Siiski katavad need korpused vaid väikest osa maailma keeltest.

3.2.1 Konverentside tulemused

2023. aastal toimus 20nes Rahvusvaheline Kõnekeele Tõlke konverents (*The 20th International Conference on Spoken Language Translation* ehk *IWSLT 2023*), kus mitmed ülesanded olid seotud kõnekeele tõlkega. Üks ülesanne oli näiteks mitmekeelne SLT, milles

keskenduti teaduslike kõnede tõlkimisele inglise keelest araabia, hiina, hollandi, prantsuse, saksa, jaapani, farsi, portugali, vene ja türgi keelde. Antud ülesandes tuli teostada üksmitmele tõlget, keerulisust lisas ülesandele valdkonnaspetsiifiline terminoloogia, salvestuse eripärad (näiteks mikrofonidest tekkiv müra, hingamised) jms. Selle ülesandega saavutasid kaskaadmudelid läbivalt paremad tulemused kui otsemeetod-mudelid. 6 parimat esitust olid kõik kaskaadsüsteemid ja viiest halvemini toiminud lahendusest olid 4 esitust otsast lõpuni süsteemid [28].

Üks teine ülesanne hõlmas endas vähese ressursiga keelte kõnetõlget. 2023 aasta ülesandes olid salvestatud lausungid järgnevatel suundadel: iiri keelest inglise keelde, marati keelest hindi keelde, malta keelest inglise keelde, puštu keelest prantsuse keelde, tamašeki keelest prantsuse keelde ja ketšua keelest hispaania keelde. Suundasid ühendas märgatav andmete vähesus, mitmel suunal oli näiteks umbes 20h kõnekeele salvestusi. Kuigi üldiselt on antud ülesanne üks kõige keerulisemaid ja tulemused näitavad seda sama, siis siiski näitasid ülesande tulemused ka mitmeid positiivseid märke. Antud aastal osales rohkem meeskondi kui kunagi varem ning lähenemised olid samuti väga erinevad - alates peenhäälestatud eeltreenitud mudelite kasutamisest kuni nullist eeltreenimiseni, kuni parameetrite tõhusa peenhäälestamiseni ja kaskaadsüsteemideni, millel kõigil näib olevat teatud määral eeliseid erinevatele keelepaaridele pakkuda [28].

Standardsetel andmestikel on kaskaad- ja otsast lõpuni süsteemide jõudluse vahe viimaste aastatega kahanenud väga väikeseks. Parimatel juhtudel võib kahe lähenemise tulemuste vahe BLEU mõõdiku järgi olla mõne 1-2 punkti suurune. Kuigi otsesüsteemid on tulemuste poolest järgi jõudnud kaskaadsüsteemidele, saavad hetkel paremaid tulemusi siiski järjepidevalt kaskaadsüsteemid. 2021 aastal oli IWSLT konverentsi tõlkimise ülesande 2 parimat tulemust saadud kaskaadsüsteemidega, parim otsast lõpuni süsteem jäi üldarvestuses 3ndale kohale ning parimast süsteemist umbes 2 BLEU punkti alla (24.6 ja 22.6) [29]. Sarnane tulemus oli ka 2022 IWSLT konverentsi tõlkimise ülesandel - kaskaadsüsteemid tõid läbivalt paremaid tulemusi (kuuel juhul kaheksast esitusest). Siiski kasutatakse üha rohkem eeltreenitud mudeleid, levinud on helikodeerijad nagu wav2vec või Hubert koodri ja mBART dekoodri jaoks. 2022 aastal kasutasid kõik võidusüsteemid eeltreenitud mudeleid ning tõlkekvaliteet paranes märgatavalt võrreldes 2021 tulemustega, mil eeltreenitud mudelite kasutamine polnud lubatud [30].

Otsemudelid on hetkel veel halvemad kui kaskaadsüsteemid ka sellepärast, et andmehulkade mahud kahe süsteemi vahel on märgatavad. Kaskaadsüsteemidele sobivaid andmehulki on kordades rohkem kui otsesüsteemidele sobivaid märgendatud andmehulki [24].

4. Närvivõrgud

Närvivõrkude kasutamine on nüüdseks levinud lähenemine mitmete erinevate ülesannete lahendamiseks. Närvivõrgud suudavad genereerida tekstist kokkuvõtteid, tõlkida, luua pilte ja sünteesida näiteks kunstlikku kõne. Järgnevalt antakse ülevaade asjakohastest terminitest, levinud närvivõrgu arhitektuuridest ning kooder ja dekooder mudelitest. Viimasena kirjeldatakse automaatseid mõõdikuid, mille abil saab kõnetõlkesüsteemide tõlgete kvaliteeti hinnata.

4.1 Põhiterminid

Masinõppe valdkond on arenenud väga kiiresti ning nüüdseks on kasutusel erinevad keerukad mehhanismid. Käesolevas jaotises antakse ülevaade närvivõrkudele omastest terminitest, kontseptsioonidest ja taustainformatsioonist.

Eeltreenimine

Eeltreenimine viiakse tavaliselt läbi suurtel andmehulkadel, et mudel saaks õppida andmetes leiduvatest mustritest ja sellele vastavalt erinevate kihtide kaale kohendada. Eeltreenimine on eriti kasulik vähese ressursiga olukordades, kui ülesandele vastavaid andmeid ei ole palju. Võrreldes kõne-teksti andmestikega, on eeltreenimiseks võimalik võrdlemisi lihtsalt leida suuri hulki andmeid, näiteks suured teksti- või kõnekorpused. Eeltreenides mudelit suure hulga varieeritud andmetega parandab see mudeli kohanemisvõimet erinevate ülesannete ja andmete jaoks. Eeltreenimist saab läbi viia erinevate meetoditega, näiteks rekonstrueerimine, maski ennustamine (*mask-prediction*) ja kontrastõppega (*contrastive learning*), et saada veelgi täpsemad esitused kontekstuaalse teabe saamiseks [24].

Peenhäälestamine

Masinõppe valdkonnas on kasutusel mudelite eeltreenimine ja peenhäälestamine kindlate ülesannete jaoks. Suurel andmehulgal treenitud mudelid ei oska tihti spetsiifilisi ülesandeid lahendada, seda parandab mudeli peenhäälestamine ülesandele spetsiifilise andmehulga ja/-või sisendiga. Nii on võimalik suurest eeltreenitud mudelist, mis saab näiteks kokkuvõtete loomisega keskpäraselt hakkama, peenhäälestada mudel eraldi kokkuvõtete andmestikul ja seeläbi parandada tulemusi märgatavalt [31].

Peenhäälestamise käigus uuendatakse eeltreenitud mudeli kaalud kasutades andmestikul

juhendatud treenimist konkreetse ülesande jaoks. Tavaliselt kasutatakse selleks sadu tuhandeid märgendatud andmenäiteid. Peenhäälestamise peamine eelis on tugev jõudlus erinevatel ülesannetel, kuid selleks peab mudelit iga ülesande jaoks eraldi andmestikul treenima. Nii pole mudel tihti võimeline kvaliteetseid järeldusi ja oletusi tegema andmete kohta, mis on tavapiiridest väljas [32].

Aktivatsioonifunktsioon

Aktivatsioonifunktsiooni kasutatakse närvivõrkudes sisendsignaali muutmiseks väljundsignaaliks, mis on omakorda järgmise kihi sisendsignaal. Tehisnärvivõrgus arvutatakse sisendite korrutiste summa ja nendele vastavad kaalud ning seejärel rakendatakse aktivatsioonifunktsiooni, et saada antud kihi väljundväärtus [33].

Isetähelepanu

Vaswani et al. tutvustas 2017 aastal artiklis " Attention is all you need" [17] mehhanismi nimega isetähelepanu (*self-attention*). See on tähelepanu edasiarendus, millega arvutatakse, et kui palju tähelepanu peaks konkreetne sisend järjestuse teistele elementidele pöörama. Seda tehakse läbi kolme vektori: päring (*query*), võti (*key*) ja väärtus (*value*). Valemil 4.1 on kujutatud isetähelepanu arvutamise valem, kus Q, K, V on esitusvektorid ja $\sqrt{d_k}$ on skaleerimise faktor.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4.1}$$

Valemit kasutatakse skoori arvutamiseks, mis näitab, kui palju peaks konkreetne sisend antud järjestuse teistele elementidele tähelepanu pöörama. Päring on selle sõna esitus, mille jaoks tahetakse arvutada isetähelepanu. Võti kujutab endast jada, mis koosneb igast järjestuses olevast sõnast, ja seda kasutatakse vastendamiseks selle sõna päringuga, mille jaoks tahetakse arvutada isetähelepanu. Väärtus on iga sõna tegelik esitus jadas. Päringu ja võtme korrutamine annab arvulise hinnangu, mis näitab, kui palju kaalu saab iga väärtus (ja seega ka sellele vastav sõna) isetähelepanu vektoris [17].

Juhendatud, juhendamata ja isejuhendatud õpe

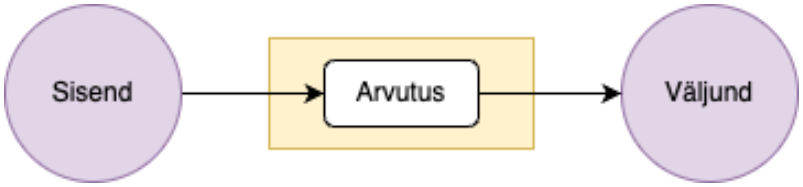
Õppe või treenimise käigus genereerib masinõppe algoritm funktsiooni, mis kaardistab sisendid soovitud väljunditega. Juhendatud õppe puhul õpib mudel andmetest, millel on olemas märgendid (*labels*). Kõnetõlke kontekstis on märgendiks näiteks helile vastav tõlge või transkriptsioon. Juhendamata õppe puhul õpib mudel andmetes olevatest mustritest, andmetel puuduvad märgendid. Pool-juhendatud õppe puhul antakse mudelile nii märgen-

ditega kui ka märgenditeta andmed. Isejuhendatud õpe on juhendamata õppe vorm, kus mudel õpib ise ennustama märgendeid olemasolevatest andmetest. Erinevalt juhendatud õppest õpib mudel enda induktiivse eelarvamuse (*inductive bias*) sisendandmetest endast [34].

Lisaks on olemas tugevdusõpe ja transduktiivne õpe. Tugevdusõppe (*reinforcement learning*) puhul õpib algoritm saades tagasisidet keskkonnalt peale igat toimingut. Transduktiivne õpe sarnaneb juhendatud õppega, kuid antud juhul on eesmärgiks ennustada konkreetse andmestiku põhjal seni nägemata andmete punkte. Erinevalt induktiivsest õppest, mille eesmärk on õppida üldist funktsiooni, keskendub transduktiivne õppimine ainult antud andmestiku uute andmepunktide ennustamisele [34].

4.1.1 Pärilevinärvivõrk

Pärilevinärvivõrk (*feed-forward neural network*) on levinud närvivõrgu arhitektuur. Pärilevivõrguks kutsutakse tsükliteta närvivõrku, kus puudub neuronite väljunditelt "tagasiside"sisendite suunas. Pärilevivõrgule omast struktuuri on kujutatud Joonisel 4. Pärilevinärvivõrgud jagunevad ühe- ja mitmekihilisteks. Mitmekihiliste pärilevivõrkude puhul on sisend- ja väljundkihi vahel vähemalt üks kiht "peidetud neuroneid". Pärilevivõrgu kihte, mille neuronid on ühenduses iga neuroniga järgmises kihis, nimetatakse täissidusateks närvivõrkudeks. Mõne ühenduse puudumisel kutsutakse neid osaliselt sidusateks [35].



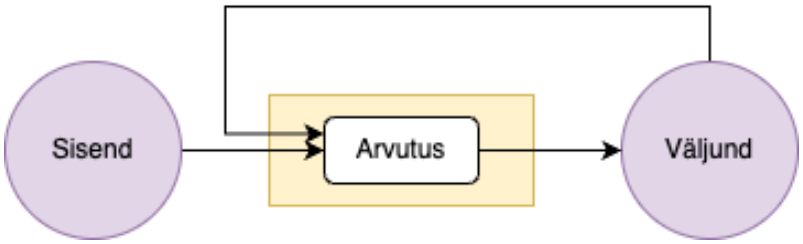
Joonis 4. Pärilevivõrk.

4.1.2 Rekurrentne närvivõrk

Erinevalt pärilevinärvivõrkudest sisaldab korduvate närvivõrkude (*recurrent neural network* ehk RNN) arhitektuur tsükleid. RNN arhitektuur on kujutatud Joonisel 5. Tsüklid annavad sisendina lisaks ka eelmiste ajaetappide teavet, et teha otsus praeguse ajaetapi sisendi kohta. Eelmise ajaetapi aktiveerimised salvestatakse närvivõrgu sisemisse olekusse ja need annavad pidevat kontekstiteavet. Pärilevivõrgus on sisendina kasutusel fikseeritud kontekstiaknad, mis katavad vaid mingi osa jadast, kuid RNN-id kasutavad dünaamiliselt

muutuvat kontekstiakent kogu jada ajaloost. Seetõttu on rekurrentsed närvivõrgud head ülesanneteks, mis hõlmavad jadade ja pidevate esituste kasutamist [36].

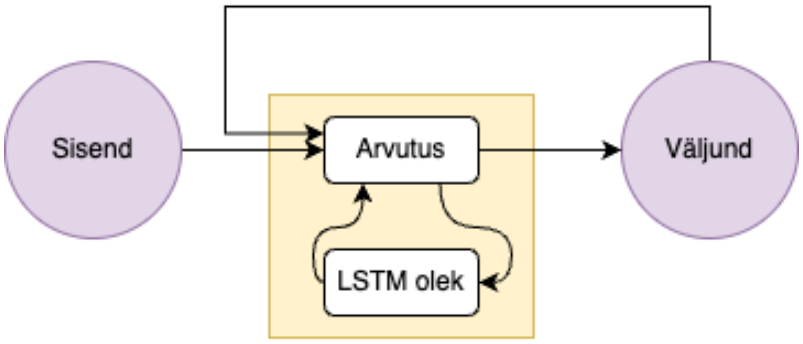
Tavapäraste RNN-ide treenimine koos gradiendipõhise tagasilevitamise (*gradient based back-propagation*) tehnikaga on siiski keeruline, sest sellega tekivad kaduvate ja plahvatuslike gradientide probleemid. Kaduvate gradientide puhul muutuvad gradiendid võrgu värskendamiste jooksul väga väikeseks, kuna need levitatakse tagasi väljundkihtidest varasemate kihtideni. Plahvatuslike gradientide probleem on sarnane, kuid gradiendid muutuvad väga suureks. Eelnevad probleemid piiravad RNN-ide võimekust modelleerida pikki kontekste, sest varasemalt nähtud asjad lähevad närvivõrgul meelest [36].



Joonis 5. Rekurrentne närvivõrk.

4.1.3 LSTM

LSTM ehk pika lühiajalise mälu (*Long Short-Term Memory*) [37] kontseptsiooni tutvustati esmakordselt 1997. aastal. See uudne rekurrentse närvivõrgu arhitektuur on loodud gradientide kadumise probleemi lahendamiseks. Gradientide kadumise probleem esineb sügavate närvivõrkude treenimisel pikkadel jadadel, mis teeb mudeli jaoks pikaajaliste sõltuvuste õppimise keeruliseks. Täpsemalt toimub see tagasilevi (*backpropagation*) protsessis, kui korrigeeritakse vigu. Gradientide põhjal korrigeeritakse mudelis kaalude väärtusi. Gradientide kadumise probleemi puhul kaugemal olevad vead ei mõjuta kaalu korrigeerimist enam piisavalt. Autorite loodud uut tüüpi rekurrentne LSTM kamber (*cell*) võimaldab mudelitel pikkade jadade puhul informatsiooni ja konteksti säilitada ning meeles pidada. Selle võtmekomponendid on mälukamber ja kolm väravat: sisendvärav, unustamisvärav ja väljundvärav. Joonisel 6 on kujutatud LSTM mudel.



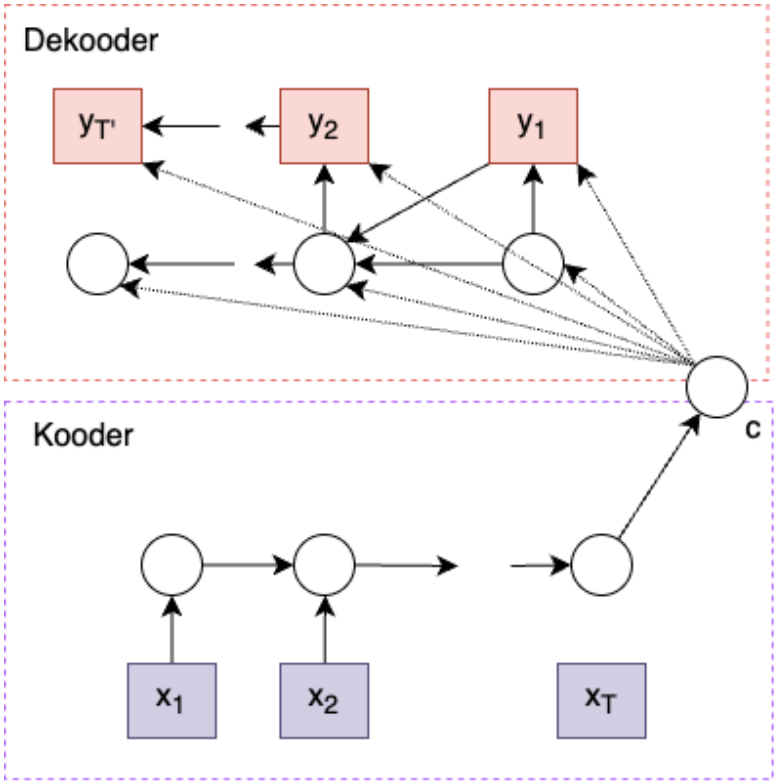
Joonis 6. LSTM mudel.

4.2 Levinud arhitektuurid

Loomuliku kõne ja keele töötluse valdkonnas on keeleliste ülesannete jaoks levinud mitmed närvivõrkude arhitektuurid. Järgnevalt tutvustatakse lühidalt erinevaid hetkel levinud närvivõrkude arhitektuure ja nendega seonduvaid mõisteid. Ülevaade antakse RNN kooder-dekooder arhitektuurist, RNN kooder-dekooder arhitektuurist koos tähelepanumehhanismiga, transformerist ning ainult dekoodrist koosnevast transformerist.

4.2.1 RNN kooder-dekooder

Kooder-dekooder arhitektuuri tutvustati esimest korda 2014. aastal. Autorid Cho et al. [15] kirjeldasid koodrit ja dekoodrit esialgu tavapärase fraasipõhise statistilise masintõlke süsteemi kontekstis ja viitasid sellele nimega *RNN Encoder-Decoder*. Sisuliselt koosneb kooder-dekooder närvivõrgumudel kahest ühiselt treenitud rekurrentsest närvivõrgust. Kooder muudab sümbolite jada fikseeritud pikkusega vektorkujule. Dekooder pöörab omakorda protsessi ümber, viies sisendi vektorkujult tagasi sümbolite jada kujule. Protsessi on kujutatud allpool Joonisel 7. Kooder loeb sisendjada x igat sümbolit järjest, peale igat lugemist rekurrentse närvivõrgu peidetud olek muutub. Peidetud olek muutub rakendades mittelineaarset aktivatsioonifunktsiooni jada eelmisele peidetud olekule ja käesolevale sümbolile. Pärast sisendjada viimase sümboli lugemist on peidetud olek c sisuliselt kogu sisendjada kokkuvõte. Mudeli teine rekurrentne närvivõrk ehk dekooder suudab väljundjada genereerida nii, et ennustab peidetud olekust järgmist sümbolit y_T.

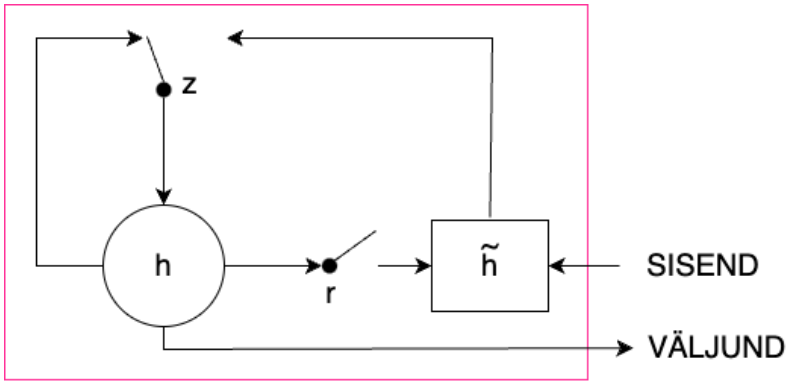


Joonis 7. Kooder-dekooder mudeli arhitektuur [15].

GRU ehk väravaga korduvüksus

Kooder-dekooder arhitektuuri tutvustavas artiklis kirjeldavad autorid esmakordselt ka teist olulist mehhanismi, mis on praeguseks laialdasemalt tuntud nimega *Gated Recurrent Units* (GRU) ehk väravaga korduvüksused. Autorid kasutavad väravaga korduvüksust koodris peidetud olekute arvutamisel aktivatsioonifunktsioonina. GRU peidetud olek on inspireeritud pika lühiajalise mälu võrgust ehk LSTMist. Võrreldes LSTMiga on artiklis kirjeldatud peidetud üksust palju lihtsam implementeerida ja arvutada. Samuti nõuab see vähem ressursse. LSTM sisaldab mälukambrit (*memory cell*) ja nelja väravat. Autorite välja pakutud varjatud üksus, mille arhitektuur on kujutatud Joonisel 8, sisaldab ainult kahte väravat: lähtestamisvärav r ja uuendamisvärav z. Igal peidetud üksusel on eraldi lähtestamis- ja uuendamisväravad. Kandidaat peidetud oleku arvutamine sõltub praegusest sisendist ja eelmisest peidetud olekust. Lähtestamisvärava r arvutatud väärtusest oleneb, kas praegune kandidaat peidetud olek $tildeh$ peab ignoreerima eelmist peidetud olekut ja võtma arvesse ainult hetkesisendit. Väärtus, mis on lähedal nullile, tähendab seda, et tuleb ignoreerida eelmist peidetud olekut ja väärtus sõltub suuresti hetkesisendist. Kõrge väärtus tähendab seda, et kandidaat peidetud olek põhineb lisaks praegusele sisendile ka eelmisel peidetud olekul. Uue peidetud oleku h arvutamisel on oluline roll ka uuendamisväraval,

mis on tähistatud z, mis üldiselt määrab, kui suur mõju on $tildeh$ uuele peidetud olekule. Seega uuendamisväravaga on võimalik kontrollida, kui palju informatsiooni eelmisest peidetud olekust uude peidetud olekusse *h* edasi kandub. [15, 38]



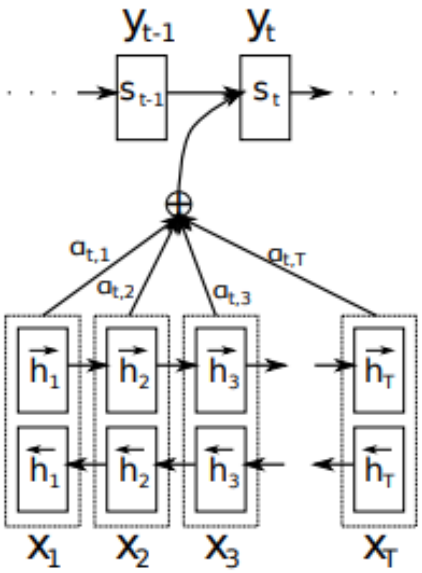
Joonis 8. GRU ehk väravaga korduvüksuse arhitektuur [15].

4.2.2 RNN kooder-dekooder tähelepanumehhanismiga

Esialgsetel neurotõlke kooder-dekooder tüüpi mudelitel oli ka teatud puudus: pikemate lausete töötlemisega mudel väga hästi hakkama ei saanud. Autorid pakkusid välja, et põhjus võis seisneda selles, et vektoresitusel, mis on fikseeritud pikkusega, pole pika ja keerulise lause hoidmiseks piisavalt võimekust. Muutuva pikkusega jada kodeerimisel võib närvivõrk mõnikord nii-öelda ohverdada sisendlauses osad olulised teemad teiste teemade meelespidamiseks [39].

Kirjeldatud probleemi lahendamiseks tegid 2015. aastal Bahdanau et al. [16] laienduse kooder-dekooder mudelile, tänu millele on mudel suuteline ühiselt sobitama ja tõlkima. Iga kord, kui mudel genereerib sihtkeelde tõlgitud sõna, otsitakse positsioone lähtesõnast, kuhu on kokku koondunud olulisim ja asjakohaseim teave. Seejärel kasutab mudel sihtkeelse sõna ennustamiseks lähtepositsioonidega seotud kontekstivektoreid koos kõigi eelnevate genereeritud sihtkeelsete sõnadega. Erinevus võrreldes tavalise kooder-dekooder mudeliga on see, et sisendlauset ei püüta kodeerida üheks fikseeritud pikkusega vektoriks, vaid hoopis vektorite jadaks. Tõlget dekodeerides valitakse adaptiivselt sellest vektorite jadast alamhulk. Seda tüüpi mehhanismi teataksegi laialdasemalt tähelepanumehhanismina (*attention mechanism*).

Pakutud mudeli tööpõhimõte on nähtav Joonisel 9, kus on kujutatud t-nda sihtkeele sõna y_t genereerimist lähtelause $(x_1, x_2, x_3, ..., x_T)$ põhjal. Välja töötatud arhitektuur koosneb koodrist, mis on sisuliselt kahesuunaline RNN, ning dekoodrist, mis emuleerib tõlke



Joonis 9. Bahdanau et al. väljatöötatud tähelepanumehhanismil põhinev mudel [16].

dekodeerimisel lähtelause läbiotsimist. Joonisel on sümbolitega $(h_1, h_2, h_3, ..., h_T)$ kujutatud annotatsioonid, mis artikli kontekstis kujutavad peidetud olekuid iga sõna jaoks. Iga annotatsiooni kaalu tähistab sümbol α_{ij}. Konkreetse sõna jaoks saadakse annotatsioon sel viisil, et ühendatakse edaspidi peidetud olek tagapoole peidetud olekuga. Sel viisil sisaldab annotatsioon h_j kokkuvõtteid eelnevatest ja järgmistest sõnadest. s_i sümbolitega tähistatakse peidetud olekut ajahetkel i. Iga annotatsioon h_i sisaldab informatsiooni kogu sisendjada kohta, aga eriline fookus on sisendjada i-ndat sõna ümbritsevatel osadel. Kogu protsess hõlmab jada iga osa jaoks kontekstivektori c_i genereerimist, mille jaoks kasutatakse annotatsioone h_i ja neile vastavaid kaale α_{ij}.

Kontekstivektori c_i arvutamine on kujutatud Valemis 4.2, milles kasutatakse h_i annotatsioonide kaalutud summa arvutamiseks väljundtõenäosusi α_{ij} ja annotatsioone h_i.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{4.2}$$

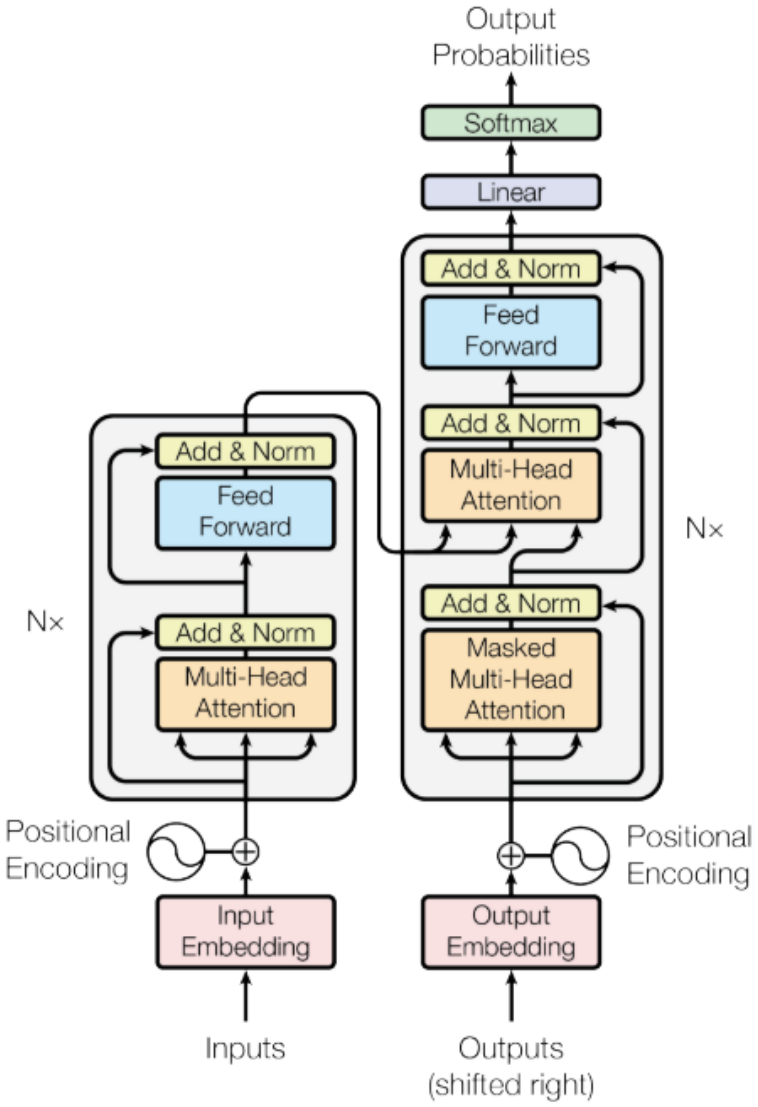
Kaalude α_{ij} arvutamine on kujutatud Valemis 4.3. Iga sõna jaoks väljundtõenäosuste normaliseerimiseks kasutatakse *softmax* funktsiooni. Kaalude arvutamise jaoks kasutatakse lisaks veel sobitamismudelit e_{ij}, mille abil hinnatakse positsiooni j ümber sisendite ja positsioonil i väljundi sobitumist. Sobitumise arvutamine on kujutatud Valemis 4.4, hinnang sõltub peidetud olekust s_{i-1} ja sisendjada j-ndast annotatsioonist.

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{4.3}$$

$$e_{ij} = a(s_{i-1}, h_j) \tag{4.4}$$

4.2.3 Transformer

Vaswani et al. tutvustas 2017 aastal artiklis " Attention is all you need" [17] arhitektuuri nimega Transformer. Transformer koosneb koodrist ja dekoodrist, kuid erinevalt teistest sellel ajal levinud arhitektuuridest kasutab transformer ka isetähelepanu ehk *self-attention* mehhanismi. Arhitektuur on kujutatud Joonisel 10.



Joonis 10. Transformer arhitektuur [17].

Transformer koosneb koodrist ja dekoodrist. Kooder koosneb kahest osast - mitmepealisest tähelepanust ja pärilevivõrgust. Dekooder koosneb kolmest osast - kahest tähelepanuplokist ja pärilevivõrgust. Dekoodri esimene mitmepealine tähelepanuplokk teostab tähelepanu mehhanismi koodri väljundi peale. Transformeri arhitektuuris on koodrid ja dekoodrid laotud, mõlemaid on 6 kihti. See tähendab seda, et ühe koodri väljundit kasutatakse järgmise koodri sisendina ning ühe dekoodri väljundit kasutatakse järgmise dekoodri sisendina [17].

Vaswani et al. [17] tõid oma artiklis mitmeid loomuliku kõne valdkonna võtmeinnovatsioone. Artiklis tutvustati isetähelepanu kontseptsiooni, mis võimaldas transformeri mudeli

luua rekurrentsete närvivõrkudeta. Lisaks kasutati mitmepealist tähelepanu ehk isetähelepanu arvutamise protsessi viidi läbi mitu korda erinevate kaalumaatriksitega. Seejärel saadi mitu vektorit ehk tähelepanupead (*attention-heads*), mis omavahel ühendati ja kaalumaatriksiga veel korrutati. Selle protsessi käigus õpib iga tähelepanupea antud jada kohta erinevat teavet, need teadmised ühendatakse lõpus. Mitmepealine tähelepanu võimaldab mudelil üheaegselt erinevate alamesituste teavet saada erinevatest positsioonidest. Luues tähelepanumehhanismi, mis võimaldab tähelepanu paralleelselt arvutada, muutus mudelite treenimine varasemast märgatavalt kiiremaks. Dekoodris on kasutusel ka maskeeritud mitmepealine tähelepanumehhanism - selleks, et ennetada teabe liikumist vasakutpidi, maskeeriti sisendi tähelepanu sees kõik väärtused negatiivse lõpmatusega, mida dekooder ei tohtinud teada enne järgmise väljundi ennustamist.

Lisaks kasutasid autorid positsioonilist kodeerimist. Kuna transformeri arhitektuur on pärilevi tüüpi, pole mudelil teavet, et mis sisendiosa paikneb millises lauseosas. Selleks lisati koodri ja dekoodri sisendvektoritele (*input embeddings*) positsiooni kodeeringud, mida arvutatakse siinus- ja koosinusfunktsioonidega [17].

4.2.4 GPT

Aastal 2018 tutvustasid OpenAI teadlased *Generative Pretrained Transformer* [40] mudelit, mida lühidalt nimetatakse GPT-ks. GPT on ainult dekoodrist koosnev transformeri arhitektuuril põhinev mudel. See mudel rakendab mitmepealist isetähelepanu toimingut kogu sisendkonteksti tookenitel, millele järgnevad positsioonipõhised edasisuunamiskihid, et luua väljundjaotus üle siht-tookenite. GPT-d eeltreenitakse suurel andmemahul ja seejärel peenhäälestatakse vastavalt ülesandele.

Transformer tüüpi mudelite jõudlus ja võimekus on aastate jooksul arenenud märkimisväärselt. Kuna kontekstipõhine õppimine hõlmab mudeli parameetrite piires mustrite omastamist, on usutav, et kontekstiga õppimise võimed tõusevad mastaabiga samamoodi märgatavalt. Transformer tüüpi mudelid on parameetrite arvult aina kasvanud - 100 miljoni parameetri pealt 300 miljonit, 1.5 miljardit, 8 miljardit, 11 miljardit ja 17 miljardit parameetrit. GPT-3 mudelil on 175 miljardit parameetrit [32]. Kuigi GPT-4 parameetrite arv ei ole avalik, võib oletada, et mudeli parameetrite arv võib olla juba triljonites.

4.3 Kooder ja dekooder mudelid

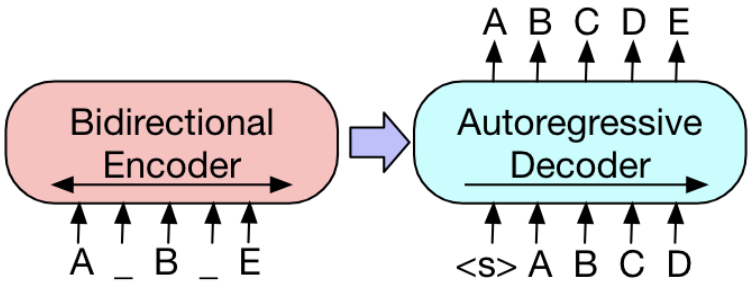
Kooder ja dekooder mudelid ning selle variatsioonid on hetkel üks enim levinud närvivõrkude arhitektuuri vorme antud valdkonnas. Sellel arhitektuuril põhinevad mitmed

laialt kasutuselolevad mudelid. Järgnevalt antakse lühiülevaade BARTi, selle mitmekeelse versiooni mBARTi ning wav2vec mudeli kohta.

4.3.1 BART

BART on 2019 aastal Facebook AI uurimisgrupi poolt välja töötatud müra eemaldav autokooder, mis on mõeldud Seq2Seq mudelite peenhäälestamiseks. BART arhitektuur ühildab kahesuunalised ning autoregressiivsed transformerid. Kahesuunaliste mudelite puhul analüüsitakse sisendi kohta tulemuse saamiseks sellele eelnevat ja järgnevat konteksti. Müra eemaldavad autokoodrid on sellist tüüpi mudelid, mida on treenitud teksti rekonstrueerimiseks sellistel puhkudel, kui mingi osa tekstist on "maskeeritud". BARTi saab kasutada mitmete erinevate ülesannete jaoks, näiteks märgi või sümboli klassifitseerimise ülesannete, jada genereerimise ülesannete (*Sequence Generation Tasks*) ning ka masintõlke jaoks.

BARTi eeltreening jaguneb kaheks etapiks. Esimeses etapis rikutakse teksti juhusliku müra funktsiooniga ning teises etapis õpetatakse Seq2Seq mudelit originaalteksti rekonstrueerima. BART mudeli arhitektuur (kujutatud Joonisel 11) on standardne transformeril põhinev neuromasintõlke arhitektuur. Jooniselt on näha, et esialgset dokumenti on rikutud, asendades osa tekstist maskeerivate sümbolitega. Seda sama rikutud dokumenti kodeeritakse kahesuunalise mudeliga. Viimaks arvutatakse algse dokumendi tõenäosus autoregressiivse dekoodriga. BART mudeli arhitektuur on sarnane Google AI keele uurimisgrupi poolt varem loodud BERT mudeli arhitektuuriga, aga esineb paar erinevust. Võrreldes BARTiga kasutab BERT enne sõnaennustust veel üht täiendavat pärilevivõrku. BART mudelis teostab dekoodri iga kiht veel ka ristsuunalist tähelepanu koodri viimase peidetud kihi üle. BART mudelil on 10% rohkem parameetreid kui sama suurusega BERT mudelil [41, 42].



Joonis 11. BART mudeli arhitektuur [41].

mBART

Multilingual BART (mBART) on BART mudeli mitmekeelne versioon, esimesena välja töötatud Liu et al. [43] poolt 2020.aastal. BART mudelit eeltreeniti vaid inglise keele jaoks. Liu et al. lõid erineval hulgal mitmekeelsetel andmetel treenitud mudelid, et hinnata ja võrrelda mitmekeelsuse mõju eeltreenimise faasis. Eeltreenimise andmed on pärit Common Crawl ehk CC andmestikust, mille hulgas on keeli erinevatest keelkondadest. Artiklis kirjeldatud erilaadsete mBART mudelite eeltreenimiseks kasutati CC andmestikust väljavõetud keelte alamhulka - CC25. Kakskümmend viis keelt, mida CC25 andmestik hõlmab, on ühekeelse korpuse suuruse põhjal järjestatult: inglise, vene, vietnami, jaapani, saksa, rumeenia, prantsuse, soome, korea, hispaania, hiina, itaalia, hollandi, araabia, türgi, hindi, tšehhi, leedu, läti, kasahhi, eesti, nepali, sinhala, gudžarati, birma. Näiteks mBART25 mudelit eeltreeniti kõigil kahekümne viiel keelel, mBART06 mudelit eeltreeniti keelte alamhulgal, mis sisaldab vaid kuute Euroopa keelt. Lause tasemel masintõlkimiseks peenhäälestati mitmekeelseid eeltreenitud mudeleid kakskeelsete tekstide ehk paralleeltekstide kogumil. Autorid rõhutasid, et mBART mudeli jõudlus paranes teatud juhtudel peenhäälestamise käigus keelte puhul, mis ei olnud osa eeltreeningu korpusest. See võib viidata sellele, et mBART on võimeline haarama universaalseid keelelisi omadusi, mis võimaldavad mudelil tulemusi saavutada ka keeltes, mida see pole otseselt õppinud. Järelikult on mBART võimeline haarama universaalseid keelelisi omadusi, leidma mustreid ja looma seoseid erinevate keelte vahel isegi, kui pole konkreetselt selle keelega varem kokku puutunud [43].

2020. aastal avaldati ka mitmekeelse BART mudeli edasiarendus - mBART-50. Tang et al. [44] suurendasid mBART poolt kaetud keelte arvu topelt, kahekümne viielt keelelt viiekümnele. Kui esialgne mBART mudel peenhäälestati kakskeelsetel tekstidel, siis mBART-50 mudeli autorid pakkusid välja eeltreenitud mudelite mitmekeelse peenhäälestamise. Mitmekeelse peenhäälestamisega saavutati märksa paremad tulemused kui varasemalt kasutatud variandiga.

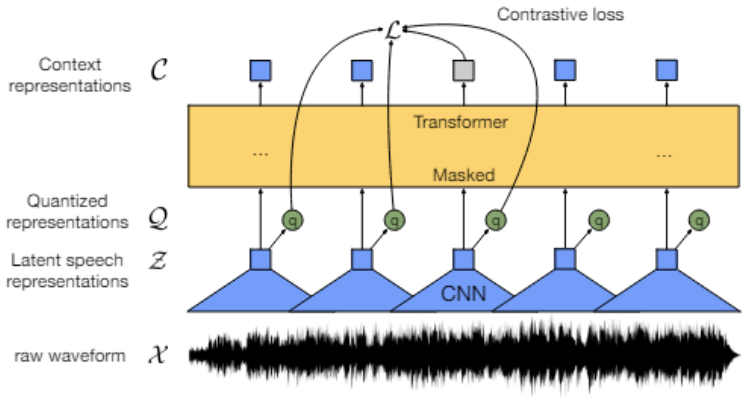
4.3.2 Wav2vec

2019. aastal avaldati wav2vec [21] mudelit tutvustav artikkel. Mudeli eesmärk oli lihtsustada märgenditeta heliandmete kasutamist. wav2vec on konvolutsiooniline närvivõrk, mille sisend on helilaine ja mis arvutab välja vektoresituse, mille saab edasi anda kõnetuvastussüsteemi. Esimene wav2vec versioon kasutas juhendamata eeltreenimist, et parandada juhendatud kõnetuvastust. See võimaldab kasutada märgenditeta heliandmeid, mida on palju lihtsam koguda kui märgenditega andmeid.

wav2vec 2.0 [22] avaldati aasta peale esmase mudeli tutvustamist. Kõneheli kodeeritakse mitmekihilise konvolutsioonilise närvivõrgu kaudu ja seejärel maskeeritakse saadud varjatud kõneesituste ulatused sarnaselt maskeeritud keele modelleerimisega. Varjatud esitused antakse transformeri mudelisse, et luua kontekstuaalseid esitusi. Mudel treenitakse kontrastse ülesandega, kus eristatakse tegelikud väärtused latentsetest segajatest. Kui varasemas wav2vec mudelis õpiti andmete kvantiseerimist, millele järgnes kontekstuaalne esitus isetähelepanu mudeliga, siis wav2vec 2.0 tegeleb mõlema osaga otsast lõpuni. Kui tolle hetkeni olid peamiselt kasutusel kaheastmelised süsteemid või mudelid, mis olid treenitud filtripanga sisendfunktsioonide rekonstrueerimise teel, siis wav2vec 2.0 autorid leidsid, et diskreetsete kõneühikute õppimine koos kontekstuaalse esitusega annab paremaid tulemusi kui etappide kaupa õppimine.

Mudel koosneb kolmest peamisest osast. Kõigepealt on funktsioonide kodeerija, mis normaliseerib ja loob lainekuju esituse. Seejärel on kontekstualiseeritud esitused transformeritega. Funktsioonide kodeerija väljund suunatakse kontekstivõrku, kus antakse esitustele asukohateabed, suhtelised positsioonid jne. Seejärel on kvantiseerimise moodul. Kõne pidev iseloom on transformerite kasutamisel üks peamisi takistusi. Kirjakeelt saab loomult diskreetselt liigendada sõnadeks või alamsõnadeks, luues sellega diskreetsete üksuste piiratud sõnavara, kuid kõnes pole selliseid loomulikke allüksusi olemas. Üks variant oleks selle süsteemina kasutada foneeme, kuid see vajaks kogu andmestiku eelnevat märgendamist - märgenditeta andmetega ei saaks enam treenida. wav2vec 2.0 õpib diskreetsed kõneühikud Gumbel-Softmaxi distributsioonist. Ühikud koosnevad koodisõnadest, mis võetakse koodirühmadest. Koodisõnad ühendatakse lõpliku kõneüksuse moodustamiseks. Seda kujutab Joonisel 12 Q ehk kvantiseeritud esitused [22].

Joonisel 12 on kujutatud wav2vec 2.0 arhitektuur. See koosneb kolmest põhiosast: konvolutsioonilised kihid, mis töötlevad töötlemata lainekuju sisendit varjatud esituse saamiseks, transformerkihid kontekstuaalse esituse loomiseks ja lineaarne projektsioon väljundi saamiseks. Mudel eeltreenitakse märgenditeta andmetel, et õppida kõnemustreid. Seejärel on võimalik mudelit täpsema ülesande jaoks peenhäälestada [22].



Joonis 12. wav2vec 2.0 arhitektuur [22].

Levinud lähenemine on näiteks mBART mudeli kasutamine ning selles oleva koodri asendamine wav2vec koodriga. See on siiani laialdaselt kasutusel olev arhitektuur, näiteks IWSLT 2023 konverentsi erinevatel võistlustel kasutasid eelmainitud lähenemist mitmed tiimid: mBART mudel, mille kooder oli wav2vec ja dekooder oli mBART50 [28].

4.4 Mõõdikud

Loomuliku kõne töötlemise valdkonnas on levinud mitmed mõõdikud. Kõige parem mõõdik tõlgete kvaliteedi hindamiseks oleks professionaalsete tõlkijate hinnangud, kuid pidevalt kõnetõlkesüsteeme arendades pole see eriti efektiivne. Käsitsi tõlkimine on aeganõudev ja kulukas protsess ning süsteeme luues on tõlkekvaliteeti vaja hinnata tihti ja kiiresti. Selleks on loodud automaatsed mõõdikud, nagu näiteks WER (*Word Error Rate*), BLEU (*Bilingual Evaluation Understudy*) ja BLEURT. Automaatsed mõõdikud võrdlevad referents-tõlget masinõppe poolt loodud kandidaat-tõlkega ja hindavad selle sobivust.

Kõnetuvastuse ja tõlkesüsteemide hindamise muudab keeruliseks ka võimalike tõlgete paljusus. Erinevatel keeltel on eri arv sõnu, sünonüüme ja võimalusi lausete konstrueerimiseks. Masinõppe jaoks võib ühel sisendlausel olla mitu väljundlauset, mis on süsteemi jaoks võrdse väärtusega, kuid inimesed on keelenüansside suhtes tundlikumad. Inimene hindab lisaks tõlke sisule ka lause loogilisust, sujuvust ja emotsiooni. Kõige paremini saaks masintõlget hinnata inimene, kuid ainult inimhindamisele tuginemine on kallis ja ajakulukas.

Transkriptsioonide hindamine on võrdlemisi lihtne, sest ASR-süsteemi hindamiseks on vaja leida kontrollitud transkriptsiooni ja masinõppe poolt antud transkriptsiooni erinevus.

Selleks on loodud laialt levinud mõõdik WER (*word error rate*). ASR-süsteemi poolt loodud lauset võrreldakse referents-transkriptsiooniga, vigade arv arvutatakse tähemärgi muudatuste summaga. Tähemärgi muudatus võib olla tähemärgi sisestus, muutus või tähemärgi kustutus. Teisisõnu leitakse referents-transkriptsioonist võetud sõna ja ASR-süsteemi genereeritud sõna Levenshtein distants ehk kui mitu muudatust peaks sõnale tegema, et saadaks referents-sõna [45].

4.4.1 BLEU

Tõlgete hindamiseks on loomuliku keele töötlemise valdkonnas kasutusel BLEU (*Bilingual Evaluation Understudy*) mõõdik. Selle mõõdiku üheks alustalaks on täpsus, mille arvutamiseks loetakse kokku kandidaattõlkes olevate sõnade arv, mis esinevad ükskõik millises referentstõlkes. Seejärel jagatakse see arv kandidaattõlke sõnade koguarvuga. Antud lähenemisega võib tekkida probleem, kui kandidaattõlkesse on hallutsineeritud sõnu, mis esinevad lausetes suure tõenäosusega (näiteks inglise keele puhul laialt kasutusel olevad sõnad nagu *the, and*). Seetõttu peaks referentssõna pärast sobiva kandidaatsõna tuvastamist pidama ennast ammendatuks. Selleks loendatakse kõigepealt maksimaalne korduste arv, et kui palju kordi esineb üks sõna ühes referentstõlkes. Seega kasutab BLEU muudetud täpsust, mis arvutatakse järgnevalt: kõigepealt loetakse kokku maksimaalne korduste arv (kui palju kordi esineb üks sõna ühes referentstõlkes), järgmiseks piiratakse iga kandidaatsõna koguarvutust selle maksimaalse referentssõna arvuga, liidetakse need piiratud arvutused kokku ja jagatakse (piiramata) kandidaatsõnade koguarvuga [46]. Valemis 4.5 on toodud BLEU mõõdiku arvutus. BP on *brevity penalty* ehk lühiduse karistus, millega korrutatakse läbi kogu ülejäänud arvutus. N on n-grammide pikkus ja p on modifitseeritud n-grammi täpsus.

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} \frac{1}{N} \cdot \log(p_n)\right) \tag{4.5}$$

Valemis 4.6 on toodud lühiduse karistuse (*brevity penalty*) valem. Selle leidmiseks võrreldakse kandidaattõlke (c) ja referentstõlke (r) pikkuseid.

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases} \tag{4.6}$$

BLEU on loomuliku kõne töötlemise valdkonnas siiani laialdaselt kasutusel oma lihtsuse

tõttu - seda on kerge ja ressursiliselt odav arvutada.

4.4.2 BLEURT

Kui BLEU vaatab, et kui palju kattuvad n-grammid, siis lause sisu BLEU siiski hinnata ei saa. Sama asja on võimalik öelda mitmel eri viisil ning BLEU on tundlik nii sõnavaraliste kui ka semantiliste ja süntaktiliste variatsioonide suhtes. Lisaks pole BLEU eriti hea hindaja juhtudel, kui võrreldavad süsteemid on sarnase täpsustasemega.

Eelnevaid probleeme proovib lahendada BLEURT [47]. BLEURT on loomuliku keele genereerimise hindamismõõdik. See võtab aluseks eeltreenitud BERT mudeli, mida on seejärel treenitud teist korda kasutades suurt hulka sünteetilisi andmeid. Viimase sammuna peenhäälestatakse mudel inimhinnangute põhjal.

Tabelis 1 on näidatud BLEU ja BLEURT skoore näidislause jaoks. Esimeses reas on toodud referentstõlge ja lähtetekst. Esimene kandidaattõlge on sama, mis referentstõlge, et näidata mõõdikute käitumist, kui sisend on sama. Teine kandidaattõlge on sisuliselt kõige lähemal referentstõlkele ning seda näitab ka BLEURT skoor.

Tabel 1. BLEU ja BLEURT skoori tulemus näidislause peal.

Arvo Pärt is a legendary composer. *Arvo Pärt on legendaarne helilooja.*	**BLEU**	**BLEURT**
Arvo Pärt is a legendary composer.	100	.945
Arvo Pärt is a well known composer.	41.1	.805
Arvo Pärt is a famous Estonian.	43.5	.635

Autorite eesmärk oli luua mõõdik, mis suudaks ekspressiivsust ja kohanemisvõimet kombineerida. Selleks võeti mitu eesmärki: esimesena asjana pidi referentslausete hulk, millel treenitakse, olema suur ja mitmekesine, et BLEURT saaks hakkama erinevate valdkondade teemade ja ülesannetega. Teiseks pidid lausepaarid sisaldama palju erinevaid leksikaalseid, süntaktilisi ja semantilisi erinevusi. Selle eesmärk on ette näha võimalikult paljusid variatsioone, mida loomuliku kõne töötlemise süsteem võib tekitada, näiteks fraaside asendused, parafraasid või väljajätmised. Selleks kasutati lauseosade ja sõnade maskeerimist, tagasitõlkimist (*back-translation*) ja sõnade väljajätmist [47].

4.4.3 Statistiline olulisus

Kui kaks süsteemi on sarnaste tulemustega, siis on võimalik nende jõudlust võrrelda kasutades statistilist olulisust. Üks viis seda määrata on Wilcoxoni astakmärgitesti abil (*Wilcoxon signed-rank test*). Mitteparameetriline Wilcoxoni test on nime saanud sellele lähenemisele 1945.aastal aluse pannud Frank Wilcoxoni järgi [48]. Seda rakendatakse kahele üksteisest sõltuvale grupile eesmärgiga, et teada saada, kas nende keskmised väärtused erinevad teineteisest märkimisväärselt [49].

Wilcoxoni testi arvutamiseks tuleb kahe valimi jaoks esmalt arvutada sõltuvate väärtuste erinevus, mis pannakse absoluutväärtuste põhjal pingeritta. Sõltuvate väärtuste erinevuste märgierinevusi peab arvesse võtma, kuna positiivsetest ja negatiivsetest erinevustest tuleb võtta summa eraldi, vastavalt T^+ ja T^-. Peale seda arvutatakse Valemiga 4.7 teststatistik *"W"* [49].

$$W = min(T^+, T^-) \tag{4.7}$$

Kui absoluutväärtuste pingereas pole mitut sama erinevusega elementi, siis eeldatav *"W"* väärtus arvutatakse Valemiga 4.8 [49].

$$\mu = \frac{n \times (n+1)}{4} \tag{4.8}$$

Juhul kui absoluutväärtuste pingereas esineb mitu sama erinevusega elemente, siis on arvutusprotsess natuke erinev. Pingereast väärtuste järjestamisel muutub see, et sama suurte erinevustega väärtustest võetakse aritmeetiline keskmine ja kasutatakse seda pingereas lõppjärjestuses nende hinnanguna. Oluline on siinkohal märkida, et algsed erinevuste märgierinevused tuleb säilitada. Seejärel saab samuti arvutada positiivsete ja negatiivsete erinevuste summa eraldi ja teststatistiku *"W"*. kasutatakse väärtust *"W"* ehk statistikat z-skoori $z = \frac{W-\mu}{\sigma}$ ja standardhälbe arvutamiseks [49].

Kõige viimase sammuna saab välja arvutada statistilise näitaja - p-väärtuse, mis näitab statistilist olulisust. See aitab kujundada hinnangut, kas tulemuste erinevused on olulise mõjuga või mitte. P-väärtuse arvutamise üks osa on hüpoteesi testimine. Selle jaoks tuleb püstitada statistiline nullhüpotees, et kaks võrreldavat gruppi on sisuliselt samaväärsed. Samuti on olemas α tase (olulisustase), millega tähistatakse statistilise olulise piirväärtust. Piirväärtuseks on sageli seatud α = 0.05. Kui p-väärtus on sellest tasemest madalam,

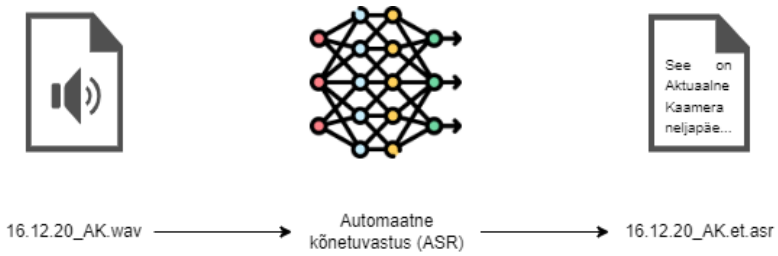
peetakse tulemust statistiliselt oluliseks. See näitab, et vaadeldava erinevuse puhul on ebatõenäoline, et see tuleneb juhuslikkusest [50].

5. Kaskaad- ja otsemeetod

Kõne tõlkimise uurimisvaldkonnas on levinud mitu lähenemist. Laias laastus jagunetakse kaheks: kaskaadsüsteemid ja otsemeetod-süsteemid. Kaskaadsüsteemide puhul on komponendid üksteise järele aheldatud, üks komponent tegeleb kõnest teksti transkribeerimisega, teine komponent saab antud teksti sisendiks ja tõlgib selle väljundkeelde. Otsesüsteemide puhul tõlgitakse sisendkõne otse sihtkeelseks tekstiks, vahesamm transkriptsiooni väljastamiseks puudub. Arvatakse, et otsemeetod-süsteemid võivad paremini töötada mitmekeelse kõne puhul ning täpsemini aru saada heli ja tõlke seostest, keelenüanssidest ning kõne eripäradest.

5.1 Transkribeerimine ja automaatse kõnetuvastuse süsteemid

Kaskaadsüsteemide üks oluline osa on automaatse kõnetuvastuse süsteem, mis muudab heli tekstiks ehk transkribeerib. Varasemalt tugineti automaatse kõnetuvastuse ehk ASR-süsteemidega eraldi komponentidele. Mõningad komponendid, millest ASR sõltus, olid keelemudel, hääldussõnastik ja akustiline mudel. ASR- ja masintõlkesüsteemide rakendamine muutus märkimisväärselt lihtsamaks tänu närvivõrkude arengule, tuues kaasa mitmeid avatud lähtekoodiga tööriistakomplekte mõlema ülesande jaoks. Sageli väljastavad ASR-süsteemid mürarikkaid transkriptsioone, kus puuduvad suurtähed ja kirjavahemärgid. See raskendab ülesannet masintõlkesüsteemi jaoks [51]. Antud töö kontekstis kasutatud automaatse kõnetuvastuse süsteemi näide on toodud Joonisel 13.



Joonis 13. Automaatne kõnetuvastus ehk ASR.

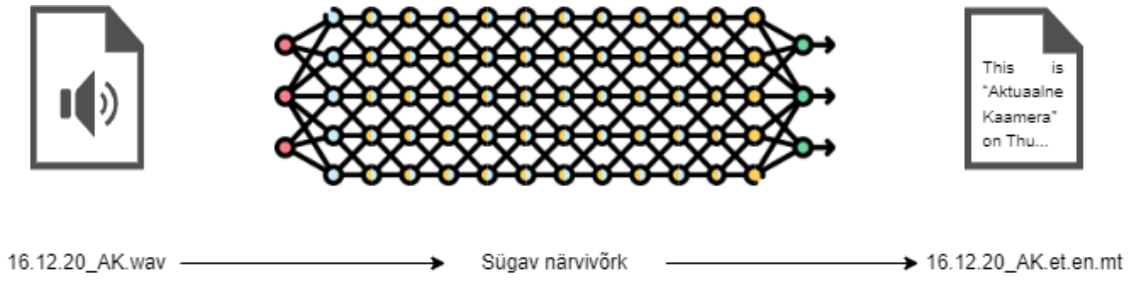
5.2 Kaskaadsüsteem

Üks laialdaselt kasutatud lähenemisviis kõnetõlkeks on kaskaadsüsteem. See saab koosneda erinevatest alamsüsteemidest, nagu näiteks kõnetuvastus (ASR) ja masintõlkesüsteem.

Kaskaadsüsteemide peamine eelis seisneb võimes kasutada ära igas alamsüsteemis tehtud edusamme, nagu näiteks viimastel aastatel välja antud suuremahulised mitmekeelsed tekstist-teksti tõlkimise mudelid ja nõrgalt kontrollitud ASR mudelid [3, 4].

Kaskaadsüsteemidel on omad piirangud, näiteks kõneteabe (sealhulgas prosoodia) kadumine. Kaduma läinud kõneteave võib olla oluline esialgse mõtte edasiandmiseks lõplikus tõlkes [3]. Samuti võib puuduseks lugeda, et näiteks ühe suuremahulise tekstist-teksti tõlkemudeliga võrreldes ei pruugi kaheastmelise (kõnest-teksti) kaskaadsüsteemi tõlked küündida sama heale tasemele. Seda tulemuslikkuse langust võib seostada mitmete nüanssidega, nagu kehvad transkriptsioonid keeltele peale inglise keele, vigade levimine ASR-mudelilt tõlkemudelile ning andmetes käsitletavate valdkondade ebakõlad eraldi treenitud alamsüsteemide vahel. Näiteks kasutades Wikipedia andmestikul treenitud ASR-mudelit koos vestluste jaoks optimeeritud tekstist-teksti tõlkemudeliga, võib see kooslus põhjustada tekstist-teksti tõlkimise etapis sisulise ebakõla. Lisaks võib teksti ületähtsustamine kaskaadsüsteemides jätta välja olulisi keelelisi tunnuseid [4].

Joonisel 14 on toodud kaskaadsüsteemi näide antud töö kontekstis. Süsteemi antakse sisse saate "Aktuaalne kaamera" helifail, ASR transkribeerib helifaili tekstifailiks, tekstifail antakse tekstist-teksti tõlkemudelile, mis omakorda väljastab sihtkeelse tõlke.



Joonis 14. Kaskaad-arhitektuur kõnest-teksti tõlke ülesande näitel.

5.3 Otsesüsteem

Otsast lõpuni lähenemine on kõne tõlkimiseks üha populaarsem uurimisvaldkond. Otsesüsteemid seavad otseselt vastavusse helisignaali ja oodatava tekstiväljundi. Otsesüsteemide tekkele on aluse pannud erinevad autorid, 2016. aastal Duong et al. [52], samal aastal ka Berard et al. [25] ja näiteks 2018. aastal Bansal et al. [53]. Kõigis kolmes töös tehti katseid kõnetõlkemudelitega, mis ei kasuta tõlkimiseks eraldi ASR-ga transkribeerimise vahesammu [54]. Esimeste seas katsetasid 2017. aastal Weiss et al. [26] mitmeülesandelist treeningut ja eelõpet otsemeetodite kontekstis, eesmärgiga kasutada täiendavaid ASR ja

masintõlke andmeid. Esimesed välja töötatud otsemeetod-mudelid olid kooder-dekooder arhitektuuril põhinevad rekurrentse närvivõrgu mudelid [3, 55]. Esialgsetest mudelitest on tehtud ka võimekamad transformer tüüpi edasiarendused, näiteks 2019 aastal Gangi et al. [55] poolt.

Otsast lõpuni kõne-teksti tõlkemudelid tõlgivad lähtekeelse kõne otse sihtkeelsesse teksti. Otsemudelid ei hõlma endas kaskaadmudelitele omaseid vahe-etappe ja esitusi, kus lähtekeelsetel andmetel treenitud ASR-süsteem on ühendatud masintõlkesüsteemiga, mis on treenitud tõlkima lähtekeelest sihtkeelde. Otsesüsteemide puhul treenitakse sageli kogu süsteem ühtse tervikuna. Treenides korraga tervet süsteemi on võimalik vältida kaskaadmudelile omaseid probleeme, kus vead ASR-süsteemist kanduvad edasi ning võivad negatiivselt mõjutada lõpptõlget. Täiendavalt on esile toodud, et otsemeetodil põhinevatest mudelitest on kasu väheste ressurssidega keelte puhul, kuna nende puhul on võimalik kasutada andmestikke, kus heli on ühes keeles ja vastav transkriptsioon teises keeles. Otsemeetodid aitavad vähendada järelduste või tulemuste viiteaega ning need mudelid on tavaliselt kontseptuaalselt lihtsamad [3, 26].

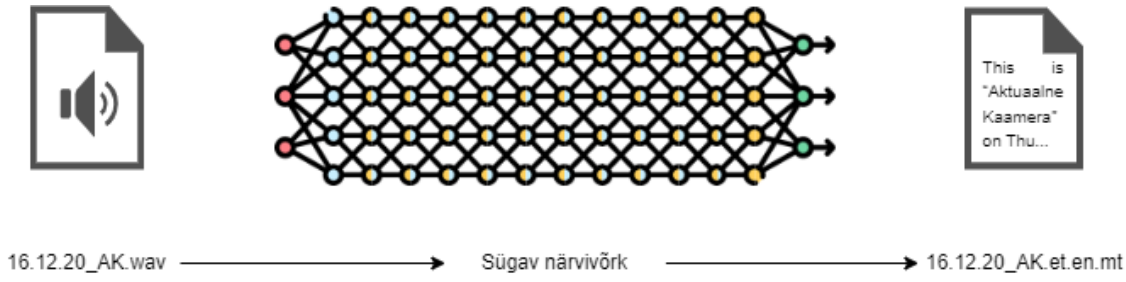
Paljud tänapäeval olemasolevad mudelid kasutavad avatud mitmekeelseid kõnekorpuseid, nagu MuST-C, EuroParl-ST, CoVoST2 ja VoxPopuli. Otsast lõpuni mudelite edusammud on märkimisväärsed, need on saavutanud akadeemilistes hindamistes kaskaadmudelitega võrdväärseid tulemusi mitmes kontekstis (näiteks piiratud andmed, valdkonnapõhised sätted, spetsiifilised keelepaarid ja nii edasi) [4]. Eelnevast on antud ülevaade ka alapeatükis 3.2.1. Samuti on otsesüsteemide eeliseks see, et nendega on võimalik säilitada keelele omaseid tunnuseid, näiteks prosoodianähtuseid, mis võivad tõlke kvaliteedile positiivselt mõjuda [3].

Otsesüsteemide eelisena on välja toodud, et seda on võimalik rakendada keeltel, millel puudub kirjakeel või mis on väljasuremisohus. Sellistel juhtudel ei saa kaskaadmeetodit rakendada, sest enamasti ei eksisteeri transkriptsioone automaatse kõnetuvastuse süsteemi treenimiseks. Otsemeetodit kasutades puudub transkribeerimise vaheetapp ning kõne on võimalik otse sihtkeelde tõlkida [53].

Otsast lõpuni meetodiga kõne tõlkimise puhul on oluline andmestike suurus. See on otsemeetodite suur puudus, et vajalikus mahus sobivaid treeningandmeid pädevate tõlkesüsteemide väljatöötamiseks napib [3]. Otsesüsteemid on ressursimahukad ning vajavad suurel hulgal andmeid, et õppida ära tundma keele omadusi ja mustreid. Selliseid andmestikke on samas vähe, sest kõnetõlkeandmete märgendamine on nõudlik nii ajaliselt kui ka ressursi poolest. Näiteks kõnetõlke andmestik MuST-C sisaldab umbes 400 tundi kõnet 230 000 lausungiga, kuid ASR andmestik Librispeech sisaldab 960 tundi kõnet ja

miljoneid paralleeltekste [24]. Otsemudeli treenimiseks vajamineva andmemahu probleemi leevendamiseks on mitmeid lahendusi. Levinumad neist on näiteks sünteetiliste andmete genereerimine, erinevate komponentide eeltreenimine ja multitegumõpe. Need meetodid kasutavad nõrgalt kontrollitud andmeid (transkriptsioonid ja tekstitõlge) lisaks kontrollitud andmetele (kõne-tõlke paarid) [56].

Joonisel 15 on toodud otsast lõpuni arhitektuuri näide kõnest-teksti tõlke ülesande näitel. Mudelisse antakse sisse helifail, seejärel õpib see erinevate mehhanismide abil helis sisalduvad mustrid ning väljastab helile vastava tõlget sisaldava tekstifaili.



Joonis 15. Otsast lõpuni arhitektuur kõnest-teksti tõlke ülesande näitel.

6. Eeltreenitud mudelid

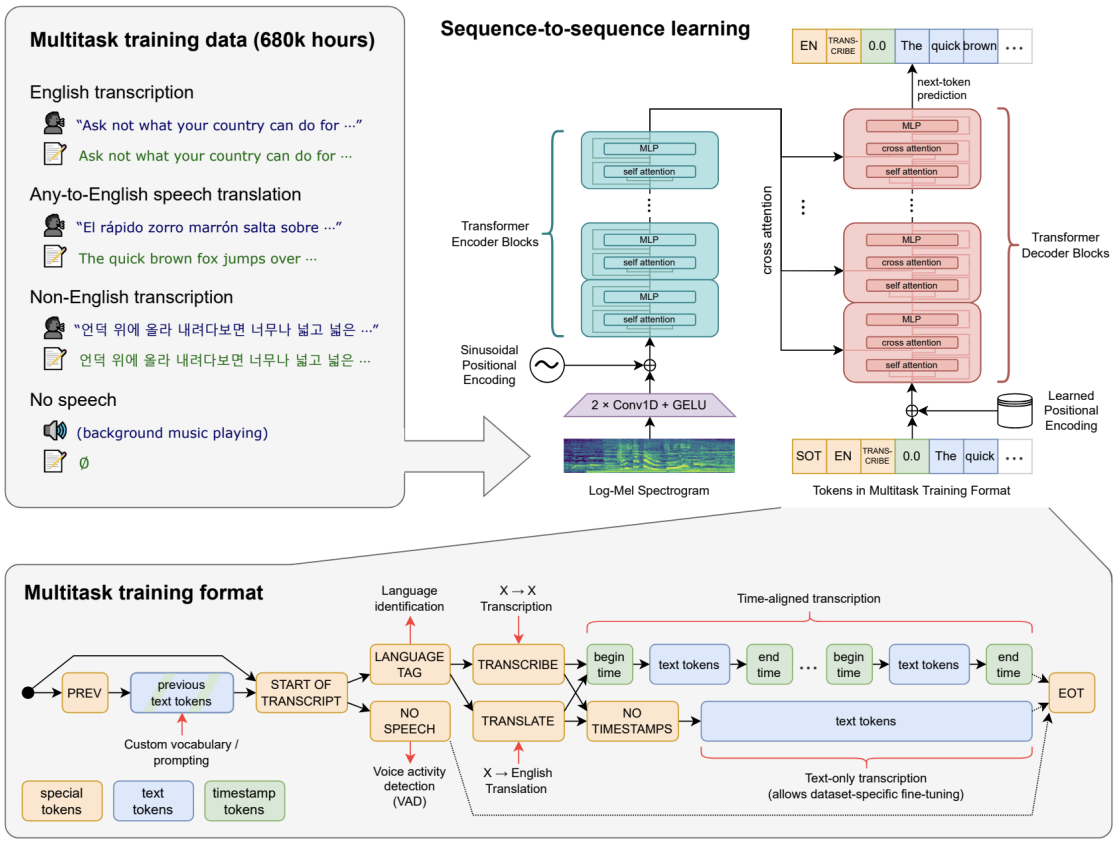
Keele ja kõne töötlemise puhul on oluline, et mudel oleks treenitud suurel hulgal andmetel. See aitab mudelil paremini mõista keeles sisalduvaid mustreid. Üldiselt ei ole tänapäeval enam mõistlik mudeleid otsast lõpuni ise treenida, sest treenimiseks vajaminevaid suuri andmehulki pole võimalik ise luua ja treenimine on ressursiliselt intensiivne. Saadaval olevad mudelid on samuti väga hea tasemega. Seetõttu on parem kasutada eeltreenitud mudeleid ning neid vastavalt ülesandele peenhäälestada. Järgnevalt antakse ülevaade erinevatest eeltreenitud masintõlke ja kõnetuvastuse süsteemidest.

6.1 Whisper

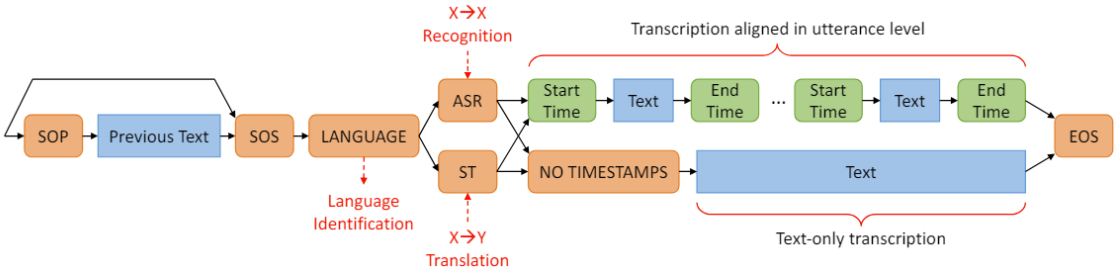
Whisper on OpenAI poolt loodud automaatse kõnetuvastuse (ASR) süsteem. See on treenitud veebist kogutud ligi 680 000 tunni mitmekeelsetel andmetel. Süsteemi loojate sõnul aitab antud suur ja mitmekülgne andmekogum parandada süsteemi töökindlust aktsentide, taustamüra ja tehnilise keele vastu. Transkribeerida on võimalik mitmes keeles. OpenAI Whisper mudel suudab ka tõlkida inglise keelde kõigist toetatud kõnetuvastuskeeltest. Muid tõlkesuundi see mudel ei toeta. Whisperi mudelid on avalikult kättesaadavad [57].

Whisperi arhitektuur, mis on kujutatud Joonisel 16, kasutab otsast lõpuni lähenemist, mis on implementeeritud kooder-dekooder transformerina. Sisendheli jagatakse 30-sekundilisteks segmentideks, teisendatakse log-Mel spektrogrammiks ja seejärel edastatakse koodrile. Dekooder on treenitud ennustama vastavat teksti (*caption*). Tekst on segatud spetsiaalsete märkidega, mis suunavad mudelit täitma selliseid ülesandeid nagu keeletuvastus, fraasitaseme ajatemplid, mitmekeelne kõnetranskriptsioon ja inglisekeelne kõnetõlge. Umbes kolmandik Whisperi andmestikust on mitte inglisekeelne. Andmetele rakendatakse vaheldumisi lähtekeeles transkribeerimise või inglise keelde tõlkimise ülesannet [57].

Whisperil on mitu eri suuruses mudelit, näiteks keskmisel Whisper (whisper-medium) mudelil on 769 miljonit parameetrit, suurel Whisper mudelil (whisper-large) on 1550 miljonit parameetrit. Whisper-large mudelil on olemas ka v2 ja v3 mudelid, mida on treenitud kauem kui esmast. Whisper mudel eeltreeniti algul juhendatult väiksemal hulgal andmetel ning seejärel suurel hulgal mürastel andmetel, kus mudel õppis ise mustreid ära tundma [57].



Joonis 16. Whisperi arhitektuur [57].



Joonis 17. OWSM arhitektuur [58].

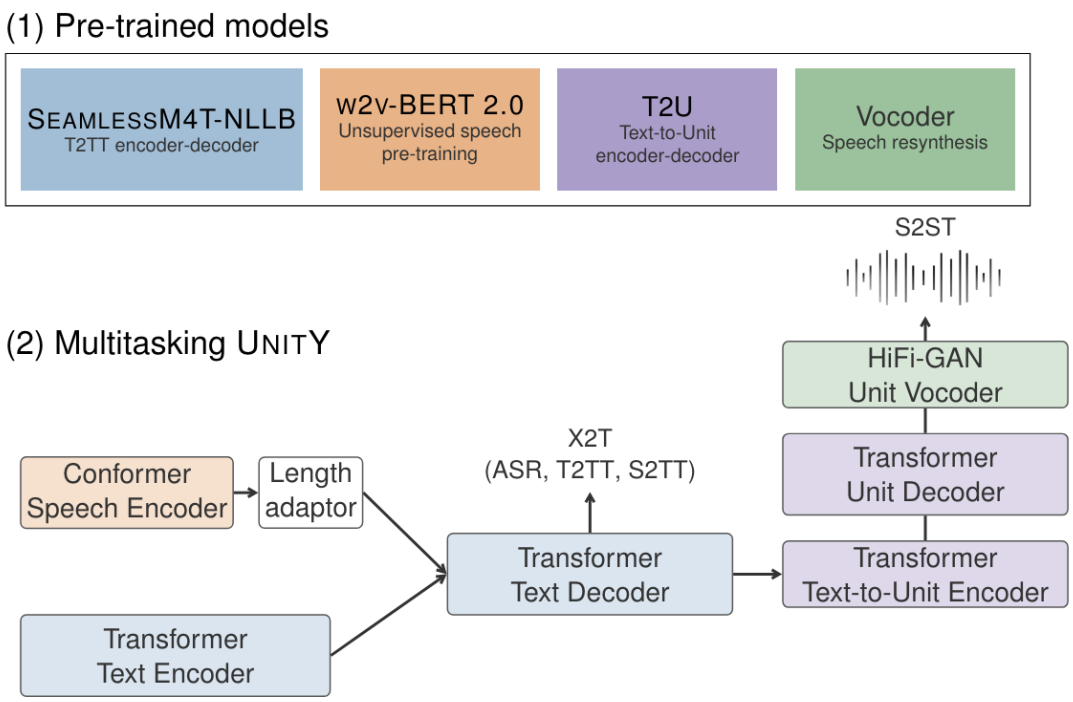
6.2 OWSM

Whisper pole avatud lähtekoodiga, näiteks pole uurijatel ligipääsu kasutusel olnud treeningandmestikule, andmete eeltöötlus ja järeltöötlus töövoogudele. Seetõttu on loodud mitmeid valmismudeleid, mis jäljendavad Whisperit. Üks selline mudel on OWSM (*Open Whisper-style Speech Model*) [58], mis on loodud 2023. aastal avalikult saadaval olevate andmestike ja ESPneti teegi abil. OWSM treeniti 180 tuhande tunni materjali peal, mis on mitu korda vähem kui Whisperi treeningandmestik. Seetõttu on autorite poolt saavutatud tulemused märkimisväärsed - need sarnanesid Whisperi tulemustele ning mõningatel näitajatel isegi ületasid Whisperi tulemusi kasutades mitu korda väiksemat treeningandmestikku. Mudeli arhitektuur on kujutatud Joonisel 17.

2024. aastal tegid Peng et al. OWSMist edasiarenduse OWSM 3.1 [59], mille eesmärk on parandada jõudlust ja tõhusust täiendavate treeningandmeteta. Esimesest kuni kolmanda versioonini (v1-v3) OWSM mudelid (ning ka Whisper) põhinevad standardsel transformer kooder-dekooder arhitektuuril. Artiklis kirjeldatud uusima OWSM versiooni kooder on väljavahetatud E-Branchformeri ehk täiustatud Branchformeri vastu. OWSM 3.1 mudelil on kaks versiooni - saja miljoni parameetriga baasversioon ja ühe miljardi parameetriga OWSM 3.1 *medium* ehk keskmine versioon. Autorite tehtud katsed näitasid, et OWSM 3.1 mudel edestab erinevates katsetes enamikel kordadel eelnevat OWSM 3 mudelit. Samuti oli uue mudeli baas- ja keskmise versiooni väljastamise kiirus Whisperi vastavatest variantidest kiirem.

6.3 SeamlessM4T

Eksisteerivatel kõnetõlkesüsteemidel on toodud välja kolm peamist puudust. Keskendutakse peamiselt kõrgema ressursiga keeltele, näiteks inglise ja prantsuse keelele. ning seetõttu jäävad väiksemad keeled katmata. Rõhku pannakse peamiselt suunale lähtekeel-inglise, mitte suunale inglise-sihtkeel. Enamik kõnest-kõnesse süsteemid toetuvad tugevalt kaskaadsüsteemile, mille sees tõlgitakse järk-järgult. Eelnimetatud puudujääkide leevendamiseks



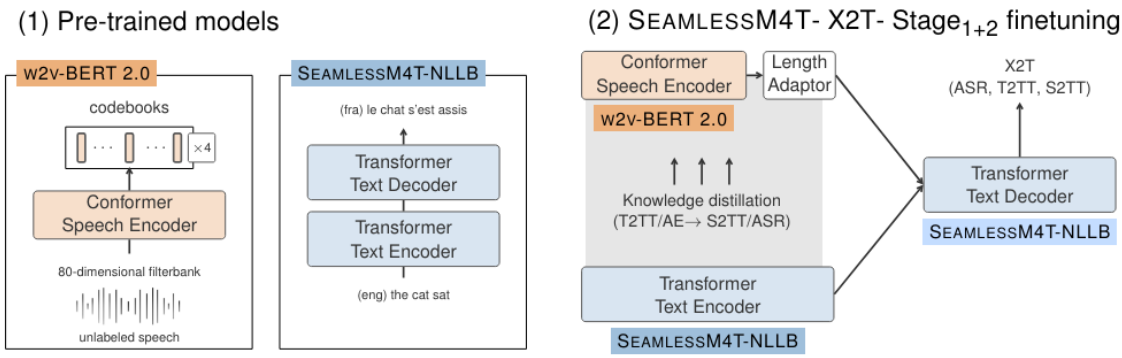
Joonis 18. SeamlessM4T ülevaade [4].

loodi SeamlessM4T [4] mudel. See on 2023. aastal Facebooki poolt loodud ühtne süsteem, mille eesmärk on pakkuda kvaliteetset tõlget ning edendada eelkõige kõnetõlke tehnoloogiaid. Autorid soovivad lisaks veel vähendada lõhet otsemudelite ja kaskaadmudelite vahel kõnest-teksti tõlkimise puhul.

SeamlessM4T mudeli arhitektuur on olemuselt otsast lõpuni, mille erinevad osad on eeltreenitud. SeamlessM4T nimes M4T tähendab massiivselt mitmekeelne ja multimodaalne masintõlge (*Massively Multilingual & Multimodal Machine Translation*). Mudel toetab kõnest-kõnesse tõlget (S2ST), kõnest-teksti tõlget (S2TT), tekstist-teksti tõlget (T2TT) ja automaatset kõnetuvastust (ASR) kuni 100 keele jaoks. Mudel hõlmab 101 keelt kõnesisestuseks, 96 keelt teksti sisestamiseks/väljastamiseks, 35 keelt kõne väljundiks [4].

Terve SeamlessM4T süsteemi ülevaade on kujutatud Joonisel 18. Mudelil on neli alustala. Esimene on SeamlessM4T-NLLB ehk tekstist-teksti tõlke kooder-dekooder mudel, mis treeniti No Language Left Behind andmestikul. Teine on w2v-BERT 2.0 kõneesituste mudel märgendamata heliandmete jaoks, selle jaoks kasutati 1 miljon tundi avatud lähtekoodiga kõne andmeid, et õppida w2v-BERT 2.0-ga isejuhendatud kõneesitusi. Kolmas on T2U ehk tekst-üksuse kooder-dekooder mudel. Neljas on vokooder, mida kasutatakse kõnest-kõneks tõlke väljundkõne sünteesimiseks [4].

Autorid lõid kaskaadmudelitega konkureerimiseks võimeka otsesüsteemi arhitektuuriga



Joonis 19. SeamlessM4T X2T ülesehitus [4].

mudeli, mis suudab tõlkida nii teksti kui ka kõnet teksti kujule. SeamlessM4T X2T ehk teksti tõlkimise ja transkribeerimise süsteemi arhitektuur on kujutatud Joonisel 19. Vasakul pool on kujutatud eeltreenitud mudelid ehk kaks peamist mudeli osa. Nendeks on w2v-BERT 2.0 ja SeamlessM4T-NLLB. Paremal on kujutatud X2T mudeli treenimist kahe faasi vältel. Esimene faas sisaldab mudeli treenimist X-inglise suunal. Teises faasis lisatakse ka vastupidine suund [4].

SeamlessM4T-laadsete mitmekeelsete tõlkesüsteemide arendamiseks on vaja väga palju erinevas vormis ressursse mitmete keelte kohta, mida sageli osade keelte ja erinevate teemade kohta napib. Töö käigus valmis ka SeamlessAlign, mis oli autorite artikli avaldamise hetkel suurim multimodaalseks tõlkeks mõeldud avatud andmestik. See sisaldab 470 tuhande tunni jagu andmeid ning hõlmab 37 keelt. Antud andmestikus on ingliskeelsest kõnest mitte-ingliskeelseks tekstiks andmeid 200 tuhande tunni jagu, mitte-ingliskeelset kõne ingliskeelseks tekstiks on 240 tuhande tunni jagu. Samuti sisaldab SeamlessAlign mitte-ingliskeelset kõne ingliskeelseks kõneks andmeid 29 tuhande tunni jagu [4].

6.4 NLLB

No Language Left Behind ehk NLLB [60] on Facebooki poolt välja töötatud tekstist-teksti tõlkemudel. Mudeli autorite eesmärk oli luua mudel, mis kataks laiemalt ka vähese ressursiga keeli, sest enamus mudeleid keskenduvad siiski laialdase kättesaadavusega keeltele. Mudel on võimeline tõlkima teksti kahesajas keeles. Mudel kasutab teksti tokeniseerimiseks SentencePiece mudelit. Sisendiks kasutati 100 miljonit lauset, kus laialdase kättesaadavusega keelte andmeid vähendati ja madala kättesaadavusega keelte andmeid suurendati, et tagada mudeli mitmekeelsus. Mudel ise põhineb transformer arhitektuuriga kooder-dekooder mudeli arhitektuuril. Kooder muudab algse tookeni jada tookeni esituseks (*embedding*). Dekooder jälgib kodeerija väljundit ja genereerib sihtlause autoregressiivselt märk-märgi haaval.

6.5 GPT mudelid

OpenAI on loonud mitu mudelit, mida hetkel kasutatakse nii teksti genereerimiseks, kokkuvõtete tegemiseks kui ka tõlkimiseks. GPT (*Generative Pre-trained Transformer*) mudeleid tutvustasid Radford et al. [40] esimest korda 2018. aastal. Arhitektuuri poolest koosnevad need mudelid vaid transformeri dekoodrist. Ainult dekoodrist koosneva transformer arhitektuuri erijuhu töötasid välja ka Liu et al. [61] 2018 aastal. Autorid leidsid, et sellise arhitektuuriga mudelil on eelis rekurrentsete närvivõrkude ja kooder-dekooder mudelite ees pikemate sisendjadade puhul. 2019. aastal avalikustati viimane avatud lähtekoodiga GPT mudel, GPT-2. Peale seda on OpenAI GPT mudelid olnud suletud lähtekoodiga [62].

6.5.1 GPT-3 ja GPT-3.5

175 miljardi parameetriga keelemudelit GPT-3 tutvustasid Brown et al. [32] esimest korda 2020. aastal. GPT-3 on OpenAI poolt välja töötatud generatiivne eeltreenitud transformer mudel. 2019. aastal loodud GPT-2 mudel ja GPT-3 jagavad sama põhimudelit ja arhitektuuri mõndade erinevustega. GPT-3 mudelil on palju rohkem parameetreid, kokku 175 miljardit. [32] Võrdluseks on GPT-2 mudelil 1.5 miljardit parameetrit [62].

GPT-3 treenimisel läksid kasutusse filtreeritud Common Crawl andmestik (kaal treeningandmestikus 60%), laiendatud WebText andmestik (kaal treeningandmestikus 22%), kaks erinevat internetipõhist raamatute korpust (kaal treeningandmestikus kokku 16%) ja ingliskeelne Vikipeedia (kaal treeningandmestikus 3%). GPT-3 mudelil täheldati ka mitmeid võimalikke parendus võimalusi, näiteks esines nõrkusi tekstisünteesis ja mitmes erinevas loomuliku keele töötlemise ülesandes. Samuti oli sellel mudelil struktuurilisi ja algoritmilisi limitatsioone, mis põhjustasid kehvema skaleeruvuse [32].

GPT-3 järeltulija GPT-3.5 kohta OpenAI tiimi poolt eraldi artiklit ilmunud ei ole. See mudel on edasiminek GPT-3 mudelilt ning on töö kirjutamise ajal laialdaselt kasutusel, hetkel kasutab seda mudelit näiteks ChatGPT tasuta versioon. ChatGPT on loomuliku keele töötlemise vestlusrobot, mida juhib generatiivne AI [63]. GPT-3.5 mudelil on olemas mitu varianti: gpt-3.5-turbo ja gpt-3.5-turbo-instruct. Neist esimene on optimeeritud töötama edasi-tagasi vestluspõhisel suhtlusel ja on kasutatav vestluse lõpetamise (*Chat Completions*) lõpp-punkti kaudu. Eelnevalt mainitud gpt-3.5-turbo-instruct mudel on aga eelkõige mõeldud juhisepõhiselt etteantud ülesande lahendamiseks. See on saadaval hoopiski *Completions* rakendusliidese lõpp-punkti kaudu. Käesoleva töö kirjutamise hetkel on konkreetne *Completions*i lõpp-punkt ja gpt-3.5-turbo-instruct mudel pärandliidese ja -süsteemi (*legacy*) staatuses [64].

6.5.2 GPT-4

GPT-4 on OpenAI tiimi poolt välja töötatud mudel, mis järgnes GPT-3.5-le. GPT-4 tehnilise raporti ilmumise hetkel, 2023. aastal, oli mudelile võimalik sisendina anda nii teksti kui ka pilte, mis polnud varasemates GPT mudelites võimalik. Tehnilises raportis toodi välja, et kuigi GPT-4 mudel on paljudes reaalse maailma stsenaariumites vähem kompetentsem kui inimene, siis on see ikkagi suuteline näitama inimtasemel sooritust paljudes ülesannetes. Mudel ületas märkimisväärselt varasemaid tipptasemel süsteeme ja olemasolevaid keelemudeleid. GPT-4 ei ole käesoleva töö kirjutamise hetkel kasutajatele tasuta kättesaadav [65].

6.6 DeepL

DeepL Translator on tõlkimismootor, mis on võimeline tõlkima kolmekümmet kahte keelt. DeepL firma kasutab oma tõlkemootoris närvivõrke ja süsteeme, mis on võimelised mõistma lausete peenemaid nüansse. Sedasi on tõlkemootor võimeline pakkuma mitmeid erinevaid tõlkevasteid lähtekeeles olevale lausele. Kasutaja saab näiteks valida vähem formaalse ja rohkem formaalse tõlkevaste ning eri sõnastuste ja lausekonstruktsioonide vahel [66].

6.7 Neurotõlge

Neurotõlge on Tartu Ülikooli keeletehnoloogia uurimisrühma TartuNLP loodud tõlkemootor. Töö kirjutamise hetkel on Neurotõlge suuteline tõlkima 29 erinevas keeles, millest 23 keelt on soome-ugri keeled. Kõik tõlkesuunad mahuvad ühe mitmekeelse tõlkemudeli sisse. Suuremad soome-ugri keeled, mille tõlkimisega Neurotõlge hakkama saab, on näiteks eesti, soome, ungari. Mudel on suuteline tõlkima ka väiksemate kõnelejate arvuga keeltes, näiteks võro, liivi, niidumari, ersa. Väheste ressurssidega keelte jaoks on mudeli treenimiseks saadaval vähe andmeid. Selliste keelte puhul on kasutatud inimeste tõlgitud lausepaare koos sünteetiliselt genereeritud lausepaaridega. Sünteetilised ehk masintõlgitud lausepaarid on sageli vigased, aga mõjuvad tõlkekvaliteedile ikkagi positiivselt. Lisaks on mudeliga võimalik tõlkida samasse keelde nagu sisend ehk esialgse lause stiili kohendada. Tõlkemootoriga on võimalik teha vigaste lausete veaparandust. Neurotõlke mudel on vabavara ja seda saab kasutada tasuta, näiteks läbi API või laadides tõlkemudeli ja skriptid alla GitHubist [67].

6.8 Google Translate API

Google Translate API on Google Cloudi poolt pakutav programmiliides, mis aitab tõlkida dokumente, veebilehti, rakendusi, helifaile ja videosid. Cloud Translation kasutab Google'i neuraalse masintõlke tehnoloogiat, mis võimaldab teksti tõlkida dünaamiliselt. Selleks kasutatakse Google'i eeltreenitud, kohandatud ja tõlkimisele spetsialiseerunud suurt keele mudelit. Mudel on saadaval põhi- ja täiustatud versioonides. Täiustatud mudel pakub kasutajale tõlke kohandamisfunktsioone, nagu näiteks valdkonnapõhine tõlge ja vormindatud dokumendi tõlge. Süsteem toetab üle 100 keelepaari tõlget [68].

7. Andmestikud

Tänapäeval on laialdaselt saadaval erinevad andmekogumid, mida saab kasutada mudelite treenimiseks ja peenhäälestamiseks. Andmestikud varieeruvad kestuse, andmetes käsitletud teemade, kõnetüübi ja muude selliste omaduste poolest.

Kõnetõlke puhul kasutatakse peamiselt kõnest-teksti andmestikke, mis sisaldavad lähtekeelset heli ja helile vastavaid sihtkeelseid tõlkeid. Lisaks on võimalik kasutada ka automaatse kõnetuvastuse andmestikke, mis sisaldavad heli ja lähtekeelset transkriptsiooni. Selliseid andmestikke on võimalik kohaldada kõnetõlke ülesande jaoks, kui andmestikus olevad tekstid tõlgitakse sihtkeelde.

7.1 Analüüsitud andmestikud

Andmestikud, mida analüüsiti, olid järgnevad: Common Voice, CoVoST ja CoVoST 2, CVSS, FLEURS, MuST-C, mTEDx, Voxlingua107, CMU Wilderness Multilingual Speech Dataset, MLS, Europarl-ST, VoxPopuli, GigaSpeech, TalTech Estonian Speech Dataset 1.0. Analüüsitud andmestikud on valdkonnas laialdaselt kasutusel, mistõttu uuriti ka nende sisulisi omadusi.

Lisaks uuriti ka mitmeid andmestikke, mis jäid lõppanalüüsist välja, kuna nende andmepunktide valdkonnad või omadused ei vastanud planeeritud nõuetele. Andmekogusid uurides keskenduti just andmestikele, mis oleksid kõige asjakohasemad kõnetõlke kontekstis.

7.1.1 Common Voice

Common Voice on Ameerika heategevusliku organisatsiooni Mozilla Foundationi projekt. See on avatud ligipääsuga mitmekeelne hääleandmekogum, mis on ühisloomena kokku pandud vabatahtlike panustajate häältest üle kogu maailma. Common Voice kõnekorpus sisaldab umbes ühe lause pikkuseid häälsalvestisi. Andmestik sisaldab andmepunktide kohta metaandmeid, mis annavad ülevaate kõnelejate soost ning vanuste jaotusest dekaadide kaupa, näiteks kahekümnendad, kolmekümnendad. Lisaks on osade keelte jaoks olemas info erinevate aktsentide kohta, näiteks inglise keele jaoks Ameerika inglise keel ning Briti inglise keel. Esindatud on erineva diktsiooni ja selgusega kõne. Korpust kirjeldava artikli avaldamise hetkeks hõlmas andmestik enam kui 50 000 inimese kõnesalvestisi

29 keeles, kokku 2500 tundi heli. Eestikeelsete andmete keskmine helisalvestise pikkus on 6.7 sekundit, ingliskeelsete andmete jaoks on keskmine helikestus 5.2 sekundit ning venekeelsed klipid on keskmiselt 5.16 sekundi pikkused [69].

Common Voice kodulehel on 30. aprill 2024 seisuga kogutud andmeid 124 erineva keele jaoks. Kokku on 31 176 tundi lindistatud heli, millest 20 409 tundi on valideeritud. Eesti keele jaoks on kogutud kokku 61 tundi andmeid 901 erineva kõneleja poolt. Andmed on 91% ulatuses valideeritud ning erinevaid lauseid on kokku 10 267. Inglise keele jaoks on sama kuupäeva seisuga häälandmeid 3485 tunni jagu 93 113 erineva kõneleja poolt, mis on valideeritud 73% ulatuses. Erinevaid lauseid on kokku üle 1.67 miljoni. Venekeelsete kõnelejate seas on enda häälega andnud panuse 3225 inimest. Häälsalvestised on 85% ulatuses valideeritud ning tundide arvestuses on heli kokku 276 tundi. Venekeelne andmestik sisaldab 47 020 erinevat lauset [70].

7.1.2 CoVoST ja CoVoST 2

CoVoST kõnetõlke andmestik on *Common Voice* kõnekõrpuse laiendus. See suuremahuline mitmekeelne kõnetõlke korpus on loodud 2020. aastal Facebook AI uurimisgrupi poolt. See hõlmab endas kõnetõlke andmeid 11 lähtekeelest inglise keelde. Andmed on pärit üle 11 000 kõnelejalt ning esindatud on enam kui 60 aktsenti. Andmestik sisaldab näiteks prantsuse, saksa, hollandi, vene, hiina ja mitmeid muid keeli. Andmestikus on enim esindatud prantsuse- ja saksakeelsed andmed. Eestikeelseid andmeid antud andmestikus ei ole [71].

CoVoST 2 andmestikus on võrreldes varasema CoVoST andmestikuga rohkem keeli: tõlkesuunal inglise-sihtkeel kokku 15 keelt ning suunal lähtekeel-inglise 21 keelt. Eesti keel on esindatud mõlemal suunal, vene keel on esindatud vaid suunal vene-inglise. Kõnelejate arv on CoVoST 2 kõnekorpuses lausa 78 tuhat. Kogu kõne kestus on viidud varasema CoVoST andmestikuga võrreldes 700 tunnilt 2880 tunnile. Eesti-inglise suunal on kõne treeningandmeid 3 tundi, inglise-eesti suunal 364 tundi [72].

7.1.3 CVSS

Common Voice based Speech-to-Speech translation corpus ehk CVSS kõnetõlke korpus põhineb peamiselt CoVoST 2 kõne-teksti tõlkekorpusel ja Common Voice kõnekorpusel. Andmestik on ainult inglisekeelne ja toetab 21 keelt. See on loodud mitmekeelse kõnest-kõne tõlkekorpuse loomiseks. Selle jaoks tuleb CVSS kombineerida Common Voice andmestikuga. Kombineerides CVSS andmestiku Common Voice andmestikuga

saab seda kasutada ka eesti-inglise ja vene-inglise suundade jaoks. Andmestiku autorid võtsid Common Voice'is sisalduvad lausungid ning tõlkisid need inglise keelde, seejärel genereeriti tõlgetest sünteetiline kõne. Autorid lõid CVSS-i raames kaks erinevat kõnest-kõneks tõlkimise andmestikku: CVSS-C ja CVSS-T. Need sisaldavad vastavalt 1872 ja 1937 tundi kõnet. CVSS-C sisaldab hea kvaliteediga ja selge kõneviisiga sünteesitud kõne. CVSS-T andmestikus on tõlgitud kõne hääl kantud üle algsest lähtekeelsest kõnest ehk igal kõnest-kõnesse tõlkepaaril on sarnased hääleomadused, hoolimata sellest, et tegu on erinevate keeltega [73].

7.1.4 FLEURS

FLEURS ehk *Few-shot Learning Evaluation of Universal Representations of Speech benchmark* andmestik on n-pidi paralleelkõne andmestik, mis sisaldab 102 keelt. FLEURSi kõneandmestiku eesmärk on laialdaselt toetada kõnetehnoloogia arengut erinevates keeltes. Selle loomisega sooviti aidata kaasa ning kiirendada vähese ressursiga keelte kõnest arusaamist. Andmestik sisaldab nii eesti, vene kui ka inglise keelt. Iga keele puhul on kõnesegmendid alla 30 sekundi pikkused. FLEURS andmestiku eestikeelne osa on 4.2 GB suurune, ingliskeelne osa on 3.76 GB ning venekeelseid andmeid on 4.25 GB jagu. Eestikeelses andmestikus sisaldab lisaks naishäältele ka meeskõnet vaid treeningandmestik. Test ja valideerimise andmekogumid sisaldavad vaid naishäälte klippe. Vene ja inglise keele jaoks on heliklippide sooline jaotuvus suhteliselt võrdne, aga mõlema keele jaoks on naishäälega klippe siiski pisut rohkem kui meessoost isikute häälsalvestisi. Eesti keele puhul on olemas helifailid ja failidele vastavad eestikeelsed transkriptsioonid. Helile vastavaid tõlkeid andmestikus eesti keelele ei ole [74].

7.1.5 MuST-C

MuST-C on 2019. aastal avaldatud mitmekeelne kõnetõlke korpus. See sisaldab andmeid kõneldava keele tõlkimiseks inglise keelest kaheksasse erinevasse sihtkeelde. Andmestik sisaldab iga sihtkeele kohta peaaegu 400 tunni jagu helisalvestisi ingliskeelsetest TED loengutest. TED loengutes käsitletakse väga erinevaid teemasid alustades teadusest ja poliitikast ning lõpetades meelelahutusega. Kaheksa sihtkeele seas, mida andmestik sisaldab, on ka vene keel. Vene keele jaoks sisaldab andmestik 2498 erinevat loengut, mis kokku teevad 489 tunni jagu transkribeeritud ja tõlgitud andmeid. Must-C andmestiku loomisel pandi rõhku sellele, et andmed oleksid võimalikult mitmekesised, hea kvaliteediga ning et rääkijad varieeruksid - mehed, naised, emakeelekõnelejad, aktsendiga kõnelejad. Autorid toovad välja, et MuST-C kõnekorpuses on iga keele jaoks artikli avaldamise hetkel tundide kohta rohkem andmeid, kui ükskõik millises teises avalikult kättesaadavas kõnekeele

tõlkimise korpuses või ressursis [7].

7.1.6 mTEDx

mTEDx ehk *Multilingual TEDx corpus* on kõnetuvastuse ja -tõlke andmestik. Mitmekeelne TEDx korpus on kokku pandud lühikestest, kuni kümne minuti pikkustest, ettevalmistatud kõnedest (TEDx loengutest) järgnevates keeltes: hispaania, prantsuse, portugali, itaalia, vene ja kreeka. Andmestikus kasutatud TEDx loengutel on ka manuaalsed transkriptsioonid ja tõlked mõndades sihtkeeltes: inglise, hispaania, prantsuse, portugali ning itaalia. Oluline on mainida, et kõik loengud ei ole tõlgitud. TEDx loengud on pärit TEDx üritustelt, mida hakati korraldama alates 2009. aastast. Need on sarnases formaadis TED loengutega. Kui üldiselt on TED loengud ainult ingliskeelsed, siis TEDx loengud saavad olla ka muudes keeltes. TEDx loenguid oli korpust tutvustava artikli ilmumise hetkeks, 2021. aastaks, tehtud enam kui 100 keeles ja salvestusi oli kokku üle 150 tuhande. Vene keele jaoks on korpuses kokku 53 tundi andmeid, mida saab kasutada automaatse kõnetuvastuse ja masin- või kõnetõlke mudelite treenimiseks [75].

7.1.7 VoxLingua107

VoxLingua107 on 2020. aastal ilmunud kõneandmete kogum, mis sisaldab andmeid enam kui 100 keele kohta. Andmestikku on sobilik kasutada kõnetuvastuse mudelite treenimiseks. VoxLingua107 andmestiku kokku panemiseks on kasutatud YouTube'i videotest välja võetud heli, mis on saadud keelespetsiifiliste otsingufraaside abil. Helifailid on segmenteeritud väiksemateks klippideks, mille pikkus jääb vahemikku kaks kuni maksimaalselt kakskümmend sekundit. Andmestikus on eesti keele jaoks kõneandmeid 38 tunni jagu, inglise keele jaoks 49 tundi ning vene keele jaoks 73 tunni ulatuses [76].

7.1.8 CMU Wilderness Multilingual Speech Dataset

CMU Wilderness Multilingual Speech Dataset on mitmekeelne kõneandmestik, mis koosneb piiblitekstidest. See hõlmab 699 keeles helifaile ja helile vastavaid tekste. Keele kohta on keskmiselt 20 tunni jagu lausepikkuseid transkriptsioone. Andmestiku autorid tõid välja, et uudisväljaanded, ringhäälingu-uudised ja näiteks Librivox.org veebilehel olevad audioraamatud tõlgitakse vaid piiratud arvul keeltesse. Seevastu sõna levitamise eesmärgil on väga paljudesse erinevatesse keeltesse tõlgitud religioosseid kirjutisi, nagu näiteks kristlik piibel ja koraan. Mõningate vähese ressursiga keelte jaoks on tehtud ka lindistusi, kuna nende keelte kõnelejate hulgas on tõenäoliselt vähem kirja- ja lugemisoskusega inimesi. Andmestik on kokku pandud bible.is veebilehel olevatest loetud uutest testamentidest [77].

Eestikeelseid andmeid antud andmestikus ei ole.

7.1.9 Multilingual LibriSpeech

Facebook AI uurimisgrupi loodud MLS ehk *Multilingual LibriSpeech* andmestik hõlmab endas kaheksas erinevas keeles loetud audioraamatute andmeid, mis pärinevad LibriVoxist. Keeled, mida see andmestik hõlmab, on järgmised: inglise, saksa, hollandi, prantsuse, hispaania, itaalia, portugali ja poola. Inglise keele jaoks on kõneandmeid kokku 44.5 tuhat tundi ning kõigi teiste keelte jaoks on kokku ligikaudu 6000 tundi. Suur mitmekeelne kõnekorpus MLS on LibriSpeech andmestiku mitmekeelne edasiarendus [78]. Algne LibriSpeech andmestik sisaldas tuhande tunni jagu loetud ingliskeelset kõne audioraamatutest [6]. Eesti keelt antud andmestik ei sisalda.

7.1.10 Europarl-ST

Europarl-ST andmestik on 2020. aastal loodud mitmekeelne kõnepõhise tõlke korpus, mis sisaldab enam kui 30 erinevat tõlkesuunda. Andmed pärinevad Eruroopa Parlamendis aastatel 2008-2012 toimunud debattidest. Keeled, millega artikli esmases väljaandes katsetusi tehti, valiti selle põhjal, et need sisaldaksid enim kõnetunde. Esmane valik sisaldas järgnevaid siht- ja lähtekeeli: inglise, saksa, prantsuse ning hispaania keel. Autorid on toonud välja, et Euroopa Parlamendis peetud kõnede ajatemplid on sageli ebakorrektsed, seega tuli andmeid mitu korda erinevate kriteeriumide põhjal filtreerida, mis vähendas esialgsete andmete hulka märkimisväärselt. Laused ja heliklipid, mida andmestik sisaldab, on maksimaalselt 20 sekundi pikkused [79]. Eesti keelt antud andmestik ei sisalda.

7.1.11 VoxPopuli

VoxPopuli on 2021. aastal Facebook AI uurimisgrupi poolt loodud suuremahuline mitmekeelne kõnekorpus. Kõnekorpus sisaldab 23 keele jaoks enam kui 400 tuhat tundi märgendamata kõneandmeid pikkusega 15-30 sekundit. Andmestik hõlmab endas 16 keele jaoks 1.8 tuhat tundi transkribeeritud kõneandmeid, lausungi maksimumpikkus on 20 sekundit. Andmestik sisaldab ka 17.3 tuhat tundi kõnest-kõneks tõlkimise andmeid. Lisaks pakub andmekogu 29 tunni jagu ingliskeelseid transkribeeritud kõneandmeid, kus kõnelevad inimesed, kes räägivad inglise keelt võõrkeelena. Viimast mainitud andmestikku on võimalik kasutada näiteks aktsendiga kõnelejate automaatse kõnetuvastusega seotud teadustöös [80].

VoxPopuli andmed on pärit 2009-2020 aastatel toimunud Euroopa Parlamendi erinevate

sündmuste ja kohtumiste salvestistest. Andmestik sisaldab ka eesti ja inglise keelt. Eesti keele kohta on märgendamata kõneandmeid 10.6 tuhande tunni jagu, transkribeeritud tunde on kokku aga ainult 3 tundi. Erinevaid kõnelejaid on transkribeeritud andmetes kokku 29. Inglise keele jaoks on märgendamata kõneandmeid 24.1 tuhat tundi, millest transkribeeritud on 543 tundi. Inglise keele puhul on kõnelejate arv oluliselt suurem, kokku on 1313 erinevat kõnelejat. Naissoost kõnelejaid on andmestikus eesti ja inglise keele jaoks vähem kui meessoost kõnelejaid, vastavalt 43.7% ja 29.6% [80].

7.1.12 GigaSpeech

GigaSpeech andmestik on 2021. aastal ilmunud ingliskeelne kõnetuvastuse korpus. Andmestik sisaldab nii loetud kui ka spontaanse kõne andmeid audioraamatutest, taskuhäälingu episoodidest ja YouTube'ist, mis on segmenteeritud lausepikkusteks osadeks. Andmetest on välja filtreeritud segmendid, millel on madala kvaliteediga transkriptsioon. Andmeid koguti 24 erinevas kategoorias, näiteks kunst, äri, krimi, ajalugu, teadus ja tehnoloogia, kultuur, reisimine jne. See sisaldab kümne tuhande tunni ulatuses transkribeeritud ja märgendatud heliandmeid ning 40 tuhande tunni jagu märgendatamata heliandmeid, mis on sobilikud kasutamiseks pool-juhendatud õppes või juhendamata õppes. GigaSpeech pakub treenimiseks viite erineva suurusega andmete alamhulka: 10h, 250h, 1000h, 2500h ja 10 tuhat tundi [81]. Eesti keelt antud andmestikus ei ole.

7.1.13 TalTech Estonian Speech Dataset 1.0

TalTech Estonian Speech Dataset 1.0 on Tallinna Tehnikaülikooli Keeletehnoloogia laboratooriumi loodud eestikeelne kõnetuvastuse andmestik. See sisaldab pikakujulisi eestikeelseid kõneandmeid ning käsitsi loodud transkriptsioone. Suurema osa materjalist on transkribeerinud mitteprofessionaalsed transkribeerijad. Andmestik sisaldab 1066 tunni ulatuses saadete helisalvestisi, näiteks saated "Aktuaalne kaamera", jutusaated, intervjuud. Rõhku on pandud sellele, et valitud saated sisaldaksid vestluskõnet. Lisaks on selles 31 tundi parlamendikõnesid ning 237 tunni jagu konverentsikõnesid, veebiseminare ja loenguid. Andmestikus on kokku 1334 tunni jagu heli koos transkriptsioonidega. Enamik eestikeelse kõne andmestiku jaoks kokku kogutud andmetest on Keeletehnoloogia laboratooriumi poolt transkribeeritud viimase 20 aasta jooksul [82, 83].

7.2 Analüüsitud andmestike ülevaade

Tabelis 2 on antud ülevaade analüüsitud andmestike omadustest. Roosa värviga on märgitud andmestikud, kus eesti keel on mingil kujul esindatud (kõne, kõne koos transkriptsioo-

nidega, kõne koos transkriptsioonide ja tõlgetega teistesse keeltesse). Tulbad võtavad kokku üldise teabe andmestiku kohta. Kogukestuse veerus on toodud kõigi andmestikus sisalduvate kõneandmete ajaline kestus. Andmestiku valdkond kirjeldab, mis teemadel või kust on pärit andmestikes sisalduvad andmed. Kõnetüübi veerg kajastab esile toodud andmestike kõnetüüpe ehk kas tegemist on loetud, spontaanse või sünteesitud kõnega. Transkriptsioonide tulp näitab ära, kas andmestik sisaldab kõnele vastavaid transkriptsioone ning paralleeltekst seda, kas on olemas ka transkriptsioonidele vastav muukeelne tekst. Viimane tulp kirjeldab konkreetselt eesti keele esindatust andmehulgas. Kui andmestik sisaldab ühtset andmehulka, on pandud eesti keele kestus selles hulgas. Kui andmestik on jagatud treenimis- (*train*), arendus- (*dev*) ja valideerimishulkadeks (*validation set*), siis on tundide arv võetud treenimisandmestiku suuruse järgi.

Tabel 2. Analüüsitud andmestike ülevaade.

Andmestik	Keeled	Kogukestus	Andmestiku valdkond	Kõnetüüp	Transkriptsioonid	Paralleeltekst	Kestus et-x / x-et
CommonVoice [69, 70]	124	31k tundi	Erinevad teemad	Loetud	Jah	Ei	61h (kõne)
CoVoST [71]	93	15k tundi	Erinevad teemad	Loetud	Jah	Ei	
CoVoST 2 [72]	22	2.9k tundi	Erinevad teemad	Loetud	Jah	Jah	3h/364h
CVSS [73]	1	1.9k tundi	Erinevad teemad	Loetud/Sünteetiline	Jah	Ei	
FLEURS [74]	102	1.4k tundi	Vikipeedia	Loetud	Jah	Jah	7.3h (kõne)
GigaSpeech [81]	1	10k tundi	Audioraamatud, podcastid, YouTube	Loetud/Spontaanne	Jah	Jah	
MuST-C [7]	9	385 tundi	TED kõned	Spontaanne	Jah	Jah	
mTEDx [75]	9	1k tundi	TED kõned	Spontaanne	Jah	Jah	
Voxlingua107 [76]	107	6.6k tundi	YouTube	Spontaanne	Ei	Ei	38h (kõne)
CMU Wilderness [77]	699	13.7k tundi	Religioon	Loetud	Jah	Jah	
MLS [78]	8	50.5k tundi	Audioraamatud	Loetud	Jah	Ei	
Europarl-ST [79]	6	500 tundi	Parlament	Spontaanne	Jah	Jah	
VoxPopuli [80]	24	400k tundi	Parlament	Spontaanne	Osaline	Osaline	3h (kõne)
TalTech Estonian Speech Dataset 1.0 [82, 83]	1	1334 tundi	Erinevad teemad	Loetud/Spontaanne	Jah	Ei	1334h

Tabel näitlikustab, kui vähe on saadaolevaid eestikeelseid kõneandmeid. Paljud andmestikud sisaldavad kas ainult kõne või kõne koos transkriptsioonidega. Kõnetõlke jaoks, mis vajab nii märgendatud heli kui ka sellele vastavaid tekstilisi tõlkeid, on valik kesine. Andmehulkade teemad on tihti spetsiifilised (religioossed tekstid, parlamendikõned jms), kõne on peamiselt dikteeritud või loetud tüüpi ja andmestikus sisalduvad helisalvestised on mõne sekundi pikkused. Kui andmestik sisaldab eesti keelt, on see üldiselt vaid loetud tundide jagu.

7.3 Otsemudelite treeningandmete lõppvalim

Otsemudelite peenhäälestamiseks vajaminevad andmed hõlmavad eri formaatides helifaile ning helile vastavate tõlgete ja transkriptsioonide kogumit. Treeningandmete valik, mille peal otsemudeleid peenhäälestada, põhines mitmel kriteeriumil. Esiteks eelistati andmeid, kus oleksid esindatud erinevad rääkijad, hääletoonid, kõnemaneerid ja keelevariandid, mis loomulikus vestluses esinevad. Teiseks jälgiti, et andmed pärineksid erinevatest valdkondadest ning kõnes käsitletavad teemad oleksid varieeritud.

Mitmekesised treeningandmed aitavad suurendada mudeli kohanemisvõimet ja tagavad, et mudel ei õpiks selgeks ainult üht konkreetset kõnemaneeri, teemat või muud sellist omadust. Samuti võib mitmekesisus parandada üldist kõnetõlke kvaliteeti, sest mudel suudab paremini kohaneda erinevate kõnelejate stiilidega.

Lisaks dikteeritud või loetud kõne andmetele otsiti ka spontaansemaid ja vähem formaalse kõnestiiliga kõneandmeid. Eesmärk oli leida andmeid, mis kataksid oma teemadega paljusid valdkondi. Juhul, kui mudelit treenida ainult loetud kõne andmetel (näiteks audioraamatute salvestustel või ette dikteeritud saadetel), ei pruugi see kuigi hästi hakkama saada igapäevaelus tavavestluses esineva sõnavara ja olukordade tõlkimisega. See on oluline, sest kõnetõlke mudel peaks suutma mõista erinevaid keelekasutuse kontekste ja olema paindlik erinevate kõnestiilide suhtes. Mudel võiks olla võimeline tõlkima nii loetud (osaliselt näiteks uudistesaated) kui ka spontaanset kõnet (näiteks intervjuud). Kokkuvõttes on võimalik tagada andmete mitmekülgsus kasutades lisaks dikteeritud ja loetud andmetele ka spontaanse või poolspontaanse kõne andmeid. See on oluline, et saavutada mudeli parem tõlkekvaliteet erinevatel tõlkesuundadel, milleks on antud juhul eesti-inglise, inglise-eesti, eesti-vene ja vene-eesti keelepaarid.

Mitmed andmestikud ei sobinud lõppvalimisse seetõttu, et seal polnud kriteeriumitele vastavaid andmeid eesti, inglise ja vene keele jaoks. Olenevalt andmestikust puudusid enamasti eesti-inglise, inglise-eesti, eesti-vene, vene-eesti tõlkesuunad. Samuti tuli osad andmestikud välistada selletõttu, et teemad ei olnud asjakohased või olid liiga spetsiifilised. Näiteks koosnesid mõned andmestikud ainult Euroopa Parlamendi debattidest või piiblitekstidest, millele on omane kindlat tüüpi kõnestiil ja sõnavara. Kõnest-teksti andmestikes, mis sisaldasid spontaanset kõne, ei olnud enamasti esindatud huvipakkuvaid tõlkesuundi. Kui mõni tõlkesuundadest eksisteeris, oli see tihti vaid ühesuunaline. Kõnest-teksti andmestikke on vähem kui näiteks automaatse kõnetuvastuse või tekstist-teksti andmestikke.

Pikakujulise spontaanse kõne jaoks mudelit treenides on oluline kasutada spontaansemat laadi kõnet sisaldavaid andmeid. Sellise kõnestiiliga andmestikke on siiski vähe, sest kõneandmete märgendamine on kallis ja ajakulukas. Seega tuli kombineerida nii dikteeritud, poolspontaanseid kui ka spontaanseid kõneandmeid. Just seetõttu, et see aitab kaasa mudeli võimekusele tõlkida näiteks vestlussaadetes, uudistes või vestlustes inimestevahelist spontaanset või poolspontaanset kõnet. Selline meedia sisaldab sageli spontaanse või poolspontaanse kõne tunnuseid: mõttepause, täitesõnu, kordusi ja kohati ka teistsugust sõnavara. Andmestike valimisel jälgiti ka helisalvestiste pikkuseid. Kuna loodav mudel on eelistatavalt pikemate lausungite tõlkimiseks, ei ole hea mudeli treenimisel kasutada rohkelt lühikesi helisalvestisi.

Kõiki kirjeldatud asjaolusid arvesse võttes läks üle vaadatud andmestikest kasutusesse vaid M suuruses alamhulk automaatse kõnetuvastuse andmestikust GigaSpeech. Andmestik tõlgiti sünteetiliselt inglise-eesti tõlkesuuna jaoks. Sünteetiline andmestik loodi ka kõikidele teistele tõlkesuundadele. Veel koguti andmeid juurde veebist. Veebiandmestik koosneb ETV+, TED, TV7, Youtube ja Amara allikatest kogutud materjalist. Veebiandmestik sisaldab erinevaid spontaanseid ja kohati ka loetud kõne andmeid.

8. Töö kirjeldus

Töö tegemiseks kasutati kahte arvutuskeskust. Enamus tööst teostati Tallinna Tehnikaülikooli Teadusarvutuste keskuse HPC *High Performance Computing Centre* klastris ning GPUdel treenimiseks kasutati Teadusarvutuste keskuse GPU serverit. Mõningaid katsetusi mudelite treenimisega tehti ka LUMI klastris, milles on võimsamad GPUd kui TalTech HPCs.

8.1 Tõlkimine

Algul tõlgiti tõlkemootoritega valideerimisandmestike referents-transkriptsioonid ning arvutati nende BLEU skoorid, et võrrelda hetkel laialdaselt kasutuselolevaid süsteeme. Selle eesmärk oli välja selgitada parim tõlkemootor erinevate tõlkesuundade andmestike tõlkimiseks.

Tõlkemootorid, millega võrdlus tehti, on järgnevad: GPT3.5-turbo, GPT3.5-turbo-instruct, GPT4, Neurotõlge, Google Translate API, NLLB-200 3.3B ja DeepL. Eelnimetatud mudelid valiti laialdase leviku ja kättesaadavuse järgi, mitmeid tõlkemootoreid on tavakasutajad harjunud igapäevaselt kasutama. Lisaks võeti valimisse ka NLLB, mis on kõnetõlke valdkonnas levinud avatud lähtekoodiga tekstitõlke mudel.

8.1.1 Tõlkemootorite võrdlus

Analüüsitud andmestike ülevaate peatükis 7.2 selgus, et kõnetõlke jaoks sobivaid andmeid on käsitletavatele tõlkesuundadele vähe või pole neid üldse. Kuna otsemeetod-mudelid vajavad peenhäälestamiseks lisaandmestikku, siis oli vaja sünteetilise andmestiku loomiseks tõlkida olemasolevaid andmeid. Selleks ülesandeks kõige sobivama tõlkemootori leidmiseks võrreldi tõlkemootoreid eestikeelsel valideerimisandmestikul. Eesmärk oli enne suures koguses failide tõlkimist välja selgitada, milline tõlkemootor on parim, et saada võmalikult hea kvaliteediga tõlked. Lisaks heale tõlkekvaliteedile oli oluline näiteks ka see, et väljastatud tõlked vastaksid reakaupa algsele tekstile, et neid saaks kõnetõlkes kasutada.

Erinevate tõlkemootorite tulemused varieerusid valideerimisandmetel suundade ja tõlkemootorite vahel, BLEU skoorid on nähtavad Tabelis 3. Siinkohal on oluline mainida, et tabelis tärniga märgistatud mudelid tõlgivad arvud sümbolitena. Näiteks teksti sees olev sõnakujul arvsõna neli on mudelilt saadavas väljundis hoopis numbri sümboli kujul

ehk 4. Eesti-inglise suunal osutus parimaks Google Translate programmiliides BLEU skooriga 38.9. Google Translate API tõlkemootorit kasutati eesti-inglise ja inglise-eesti tõlkesuundade andmete tõlkimiseks. Eesti-vene suunal saadi kõrgeim BLEU skoor 31.3 kasutades OpenAI GPT4 programmiliidest. Kuigi GPT4 osutus eesti-vene suunal referents-transkriptsioonide tõlkimisel edukaimaks ja GPT3.5 mudelil oli parim hinna ja tõlkekvaliteedi suhe, siis ei suutnud GPT mudelid alati iga kord tõlkida nii, et lausete ja teksti struktuur säiliks. Kuna oli oluline, et tõlked vastaksid rida-realt algsele tekstile, siis otsustati kasutada eesti-vene ja vene-eesti tõlkesuunal Neurotõlget.

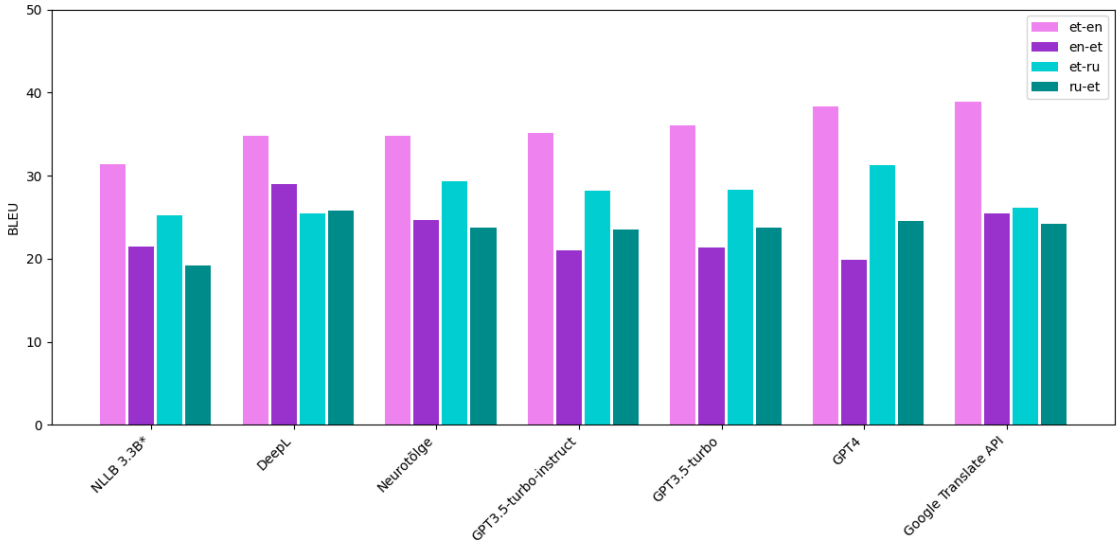
Tabel 3. Tõlkemootorite tulemuste võrdlused referents-transkriptsioonidel.

Tõlkemootor	**BLEU**					**BLEURT**				
	Eesti keelest		**Eesti keelde**			**Eesti keelest**		**Eesti keelde**		
	Inglise	**Vene**	**Inglise**	**Vene**	**avg**	**Inglise**	**Vene**	**Inglise**	**Vene**	**avg**
GPT3.5-turbo	36.1	28.3	21.3	23.8	27.4	.696	.703	.593	.665	.664
GPT3.5-turbo-instruct	35.1	28.2	21.0	23.5	27.0	.697	.711	.702	.655	.691
GPT4	38.3	31.3	19.9	24.6	28.5	.702	.721	.609	.656	.672
Google Translate API	38.9	26.1	25.4	24.2	28.7	.690	.686	.576	.655	.652
NLLB-200 3.3B*	31.4	25.2	21.5	19.2	24.3	.652	.665	.529	.574	.605
Neurotõlge	34.8	29.3	24.7	23.7	28.1	.656	.672	.558	.619	.626
DeepL	34.8	25.5	29.0	25.8	28.8	.678	.695	.605	.638	.654

Tõlkemootorite tulemuste võrdlust kujutab ka Joonis 20. Tõlke kvaliteet oleneb oluliselt tõlkesuundadest, üldiselt saavad mudelid paremini hakkama suundadega, mille lähtekeel on eesti keel. Keerulisemad on mudelite jaoks tõlkesuunad, mille sihtkeel on eesti keel. See võib tuleneda sellest, et mudeleid on treenitud suurel hulgal kõrge kättesaadavusega keelte (näiteks inglise keel) andmetel ning seetõttu on nende keelte tõlgete väljastamine mudelile loomulikum kui näiteks eestikeelsete tõlgete väljastamine.

8.1.2 Tõlkimise protsess

Tekstist-teksti tõlkimine teostati mitme erineva programmiliidesega. Tõlkemootorite võrdlus loodi selleks, et hinnata, millised mudelid oleksid parimad valikud transkribeeritud andmete tõlkimiseks ja sünteetilise andmestiku loomiseks. Valitud süsteemideks olid OpenAI, Google, Facebook, Tartu Ülikooli ja DeepL mudelid. OpenAI mudelitest olid valimis GPT3.5-turbo, GPT3.5-turbo-instruct ja GPT4 mudelid. Google mudelitest kasutati Google Translate API ja Facebooki mudelitest No Language Left Behind 3.3B parameetriga mudelit. Tartu Ülikooli puhul kasutati Neurotõlge mudelit, lisaks kasutati tõlkimiseks ka DeepL tõlkemootorit.



Joonis 20. Tõlkemootorite tulemused referents-transkriptsioonidel.

Kõikide mudelitega tõlkimiseks loodi skriptid, mis viisid andmestiku mudelile sobivale kujule, et seejärel käivitada tõlkimine. Tõlgiti ridahaaval, mistõttu oli oluline, et skriptid annaksid andmestikust ette vaid nii palju lauseid ja tähemärke kui vaja.

Programmiliideste vahel ilmnesid tõlkimise käigus mitmed huvitavad erinevused. Näiteks GPT mudelitega tõlkides oli suhteliselt keeruline saada neid tõlkima rea kaupa, mudelid tahtsid tihti sõnastada asju ümber ja kombineerida lauseid kokkuvõtvamateks uuteks lauseteks. See juhtus ka siis, kui eraldi juhendlauses öelda, et säilita paragrahvide struktuur. Üldiselt järgis mudel lausestruktuuri lühemate lausete puhul, kuid pikemate lausetega läks andmestik nihkesse. Ridade kaupa erinevate keelte vastavuses hoidmine oli oluline treenimise jaoks, kuna .vtt failides on märgitud ajavahemik, mil lause öeldakse. Kui oleks olnud vaid väikesed vahed algse ja tõlgitud teksti vahel, oleks ilmselt mingil viisil olnud võimalik need vastavusse viia, kuid erinevused olid 2-50+ rea vahemikus ning mingisugust seaduspära nende varieeruvuses ei tuvastatud. Märkimisväärne oli ka erinevate keelte tõlkimise kiiruse erinevus - inglise keele tõlkimine läks mudelitel palju kiiremini kui vene keele tõlkimine. Samuti olid GPT-3.5-* mudelid kiiremad kui GPT-4 mudel.

Neurotõlke mudeli puhul tekkis väike erinevus mudeli veebilehe ja API kasutamisel. Algselt veebilehe kaudu saadavalolevat rakendusliidest testides, et näha, kas see sobib antud kasutusjuhuga, andis veebiversioon tagasi tõlgitud teksti koos muudetud reapiiridega. Seevastu kasutades programmiliidest läbi skripti, säilisid reapiirid ja teksti struktuur. Mudel lisas vahetevahel kirjavahemärke ka kohtadesse, kus neid algtekstis polnud.

Osad mudelid tõlgivad numbrid sümboliteks ja osad sõnadeks. Antud töös käsitletud mudelitest tõlgivad peenhäälestamata SeamlessM4T v2 (large) ja NLLB-200 3.3B numbrid märkideks, kuid kasutuselolevad valideerimisandmestike referentstõlked sisaldavad numbreid välja kirjutatud sõnadena. See asjaolu võib mõjutada eelnevalt mainitud mudelite BLEU skoore.

8.2 Andmestikud

Mõned kõnetõlke andmestikud sisaldavad eesti keelt, kuid seda üldiselt vähesel määral, nagu ilmnes peatükis 7. Kui andmestik sisaldab eesti keelt, on see enamasti loetud ja/või konkreetse valdkonna kohta. Selleks, et treenida otsemeetod-mudelid paremini tõlkima tõlkesuundi, kus üks keeltest on eesti keel, on vaja suurt hulka eesti keelt sisaldavaid kõneandmeid. Selliste andmete loomiseks kasutati nii veebist täiendavate andmete kogumist kui ka andmete juurde sünteesimist.

8.2.1 Veebiandmestik

Tulenevalt eesti keele kõnelejate suhteliselt väikesest arvust, on kõnetõlkemudelite koolitamiseks veebis saadaolevate kõneandmete hulk piiratud. Eesmärk oli leida andmeid, mis sisaldavad pika vormiga vestluskõnet (mitte üksikuid lausungeid), kuna üldiselt vajavad otsemudelid paarikümne sekundilisi kõnesegmente, et treenida pika kõne transkribeerimiseks ja tõlkimiseks võimelisi mudeleid.

Töö jooksul koguti kokku erinevad internetis leiduvad videod ja helisalvestised, millele olid teoste autorid, tõlkijad või vabatahtlikud hea kvaliteediga subtiitrid koostanud. Subtiitrite kvaliteeti hinnati selle põhjal, et need oleksid heliga sünkroonis ja vastaksid enamasti sõnasõnalt lausungitele. Paljudes internetis saadaolevates videotes ja heliklippides oli subtiitrite tekst heliga võrreldes ümbersõnastatud ja esialgsest informatsioonist oli tehtud lihtsam ja lühem versioon. Subtiitreid sel viisil lühendades on tekst lõppkasutajale selgem, kuna ei sisalda üleliigseid täite- ja parasiitsõnu, mis pole olulised teksti sisust aru saamiseks. Kõnetõlkemudeli treenimisele mõjub see üldiselt siiski negatiivselt. Mudelite jaoks loodud lisaandmestikku ei kaasatud selliseid transkriptsioone. Lisaks välditi ka masintõlgitud subtiitreid.

Videote otsimise protsessis olid olulised ka teised kriteeriumid, näiteks teemade ja valdkondade mitmekesisus, rääkijate mitmekesisus ning videote pikkus. Üldiselt eelistati pikema kestusega videosid. Videod koguti kokku ja tõmmati yt-dlp teeki kasutades skriptiga alla. Seejärel töödeldi andmeid, et viia need mudelitele sobivale kujule.

Ülevaade veebist kogutud andmete hulgast iga keelepaari jaoks on näha Tabelis 4. Töö käigus leiti mitu head allikat: ETV+ (Eesti Televisiooni venekeelne telekanal), TED kõned eestikeelsete subtiitritega, TV7 (kristlik telekanal) ja erinevad Youtube kanalid, millel olid läbivalt kvaliteetsed subtiitrid. Nagu tabelist on näha, erines andmete hulk tõlkesuundade vahel palju.

Veebiandmestiku vene-eesti tõlkesuuna treeningandmete jaoks koguti 578 tunni jagu sobivaid Raadio 4 raadiosaadete salvestisi. Kuna nende helifailide jaoks transkriptsioone ja subtiitreid saadaval ei olnud, siis oli tarvis need luua. Transkribeerimiseks kasutati whisper-ctranslate2 käsurea klienti. Transkribeeritud failides olid Whisperi genereeritud ajatemplid heliklippide lausungite reaalsetest ajatemplitest liialt erinevad ning failid sisaldasid palju hallutsinatsioone. Kõiki asjaolusid arvesse võttes jäid kogutud Raadio 4 raadiosaated lõplikust valimist siiski välja.

Tabel 4. Veebiandmestik.

Allikas	**Eesti keelest**		**Eesti keelde**	
	Inglise	**Vene**	**Inglise**	**Vene**
ETV+	-	-	-	182.71h
TED	-	-	41.16h	-
TV7	-	-	16.43h	-
Youtube	39.56h	18.24h	-	433.94h
	39.56h	18.24h	57.59h	616.65h

8.2.2 Sünteetiline andmestik

Kõnetõlke jaoks sünteetiliste andmete genereerimine käib üldiselt kahel peamisel viisil. Üks variant on kasutada olemasolevat tekstist-teksti andmestikku ja sünteesida sellele lisaks kõne andmed. Teine variant on võtta andmestik, mis sisaldab transkribeeritud kõne ja tõlkida transkriptsioonid soovitud sihtkeelde. Antud töö kontekstis otsustati kasutada teist varianti, sest eksisteerivad ASR andmestikud on poolspontaanset laadi, ja olemasolevad masintõlke süsteemid tõlgivad eesti keelt suhteliselt kõrgetasemeliselt. Sünteetiline andmestik loodi iga tõlkesuuna jaoks. Eesti-inglise ja eesti-vene tõlkesuundadel kasutati sama eestikeelset andmestikku, mis tõlgiti nii inglise kui ka vene keelde. Tabelis 5 on toodud tõlkesuundadele vastavalt sünteetiliste andmete allikad, kestvused ja valdkonnad.

Eesti-inglise ja eesti-vene suuna jaoks loodi sünteetiline andmekogu tõlkides TalTechi

Eestikeelse Kõne Andmestikust 1.0 [82] pärinevad eestikeelsed transkriptsioonid. Tõlkemootoriks valiti eesti-inglise suunal parima tulemuse saavutanud Google Translate API. Eesti-vene tõlkesuunal kasutati Neurotõlget, et tõlkida eestikeelsed transkriptsioonid vene keelde. Kuna tõlgiti suurt mahtu andmeid (mõningatel suundadel 1000+ tundi), siis tuli lisada tõlkimise skriptidele ooteajad, et rakendusliideseid päringutega mitte üle koormata.

Inglise-eesti suuna jaoks läks kasutusse GigaSpeech [81] andmestiku M suuruses alamhulk (kestus 1000 tundi), mille ingliskeelsed transkriptsioonid tõlgiti Google Translate APIt kasutades eesti keelde. Selleks, et saada vene-eesti tõlkesuuna jaoks sünteesitud andmed, tõlgiti Neurotõlke tõlkemootorit kasutades TEDx kõned ja YouTube'ist Deutche Welle venekeelselt kanalilt erinevatest videotest pärinevad transkriptsioonid.

Tabel 5. Sünteetiline andmestik.

	Eesti - x	**Inglise - eesti**	**Vene - eesti**
Andmestik	Estonian Speech Dataset	GigaSpeech	Deutsche Welle ru, TEDx
Kirjeldus	Saated: 1066h Konverentsid, seminarid: 237h Parlamendikõned: 31h	Audioraamatud: 260h Taskuhäälingud: 350h Youtube: 390h	Deutsche Welle ru: 45h TEDx: 57h
Kokku	1334h	1000h	102h

8.2.3 Valideerimisandmestik

Nelja tõlkesuuna hindamise jaoks loodi kolm valideerimisandmestikku. Eesti-inglise ja eesti-vene suunal koosneb valideerimisandmestik seitsmest samast helisalvestisest, mille kestus on 4.15 tundi. Andmestikus on kolm "Aktuaalse kaamera" saadet, kaks pressikonverentsi ja kaks "Ringvaade"saadet. Salvestised on transkribeeritud ja transkriptsioonid on tõlgitud tõlkijate poolt inglise ja vene keelde. Vene-eesti ja inglise-eesti suunal koosnevad valideerimisandmestikud YouTube'st käsitsi valitud videotest. Videote kogumikud koosnevad peamiselt vestlussaadetest, uudissaadetest ja pressikonverentsidest. Nii on valideerimisandmestikus esindatud nii poolspontaanne kui ka pigem dikteeritud kõne. Näiteks vestlussaated on iseloomult vabama kõnestiiliga, seevastu uudissaadetel on suhteliselt kindel kava, mida järgitakse. Inglise-eesti suuna andmestiku suurus on 3.05h ja vene-eesti andmestiku suurus on 4.51h. Ülevaade valideerimisandmetest on toodud Tabelis 6.

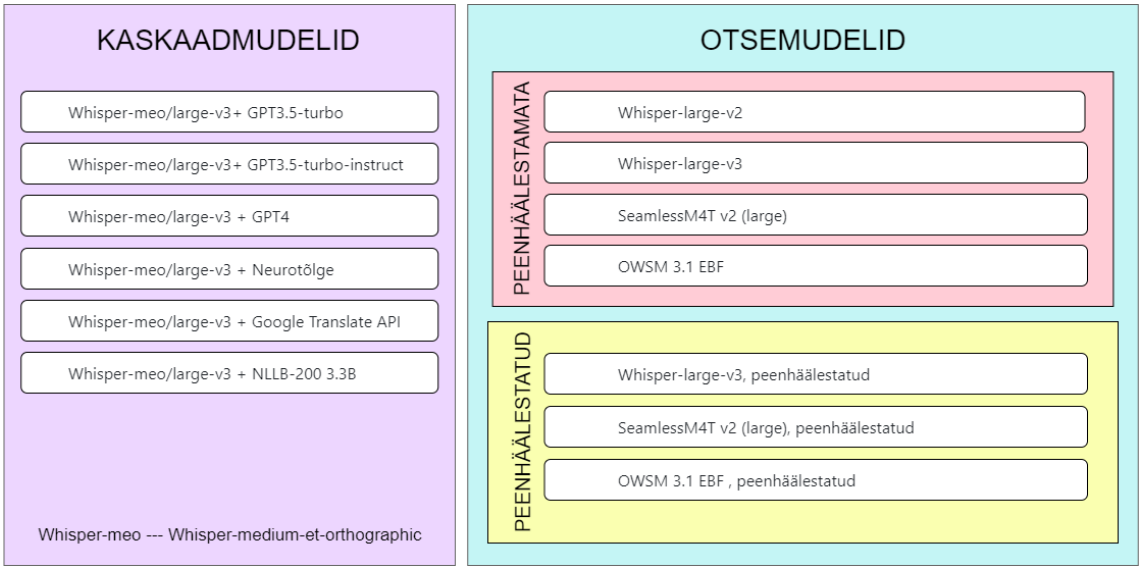
Tabel 6. Valideerimisandmestike suurus tõlkesuuna põhiselt.

Tõlkesuund	Kestus	Failid
Eesti - x	4.15h	7
Inglise - eesti	3.05h	5
Vene - eesti	4.51h	6

8.3 Mudelid

Kaskaadmudelid, millega viidi läbi katsed, kasutasid eesti-x tõlkesuunal Whisper-medium-et-orthographic automaatse kõnetuvastuse mudelit ja x-eesti tõlkesuunal Whisper-large-v3 mudelit. Whisper-medium-et-orthographic on Whisper-medium mudel, mis on peenhäälestatud umbes 800h eestikeelsetel andmetel. See on loodud Kõnetehnoloogia labori poolt Tallinna Tehnikaülikoolis. ASR-süsteemidega kombineeriti tõlkemootorid, milleks olid GPT-3.5-turbo, GPT3.5-turbo-instruct, GPT-4, Neurotõlge, Google Translate API ja NLLB-200 3.3B.

Otsemudelid, millega viidi läbi katsed, on järgnevad: Whisper-large-v2, Whisper-large-v3, SeamlessM4T v2 (large), OWSM 3.1 EBF. Whisper on OpenAI poolt loodud eeltreenitud automaatse kõnetuvastuse ja kõne tõlkimise mudel, mis on valdkonnas laialdaselt kasutusel eksperimentide läbiviimiseks. OWSM mudel valiti seetõttu, et see jäljendab Whisper stiilis mudelit. SeamlessM4T on avatud lähtekoodiga mudel, mis on loodud Meta poolt. SeamlessM4T mudel valiti sellepärast, et loomuliku kõne valdkonnas on see Whisperiga sarnasel või isegi paremal tasemel. Nii Whisper kui ka SeamlessM4T on levinud mudelid, mida kasutatakse palju. Mudelite jaotuvus kaskaad- ja otsemeetodi vahel on toodud Joonisel 21.



Joonis 21. Mudelite jagunemine.

8.3.1 Transkribeerimine ja failide töötlemine

Transkribeerimine

Klastris transkribeeriti helifailid kasutades Whisper mudeleid automaatkõnetuvastuseks. Eesti-inglise ja eesti-vene suundadel kasutati transkribeerimiseks Whisper-medium-et-orthographic mudelit ning inglise-eesti ja vene-eesti suundadel kasutati Whisper-large-v3 mudelit. Järgnevalt on näide Whisper-large-v3 kasutamisest helifailide transkribeerimiseks.

```
srun -p gpu --gres gpu:1 -t 72:00:00 --mem 64G
--cpus-per-task=8
--pty whisper-ctranslate2 --language {et, en, ru}
--prompt\_reset\_on\_temperature 1.1
--task transcribe --model large-v3
--vad\_filter True --beam\_size 5
--compute\_type float32
<vahedega eraldatud helifailide nimed>
```

Transkribeerimise tulemusel loodi erinevates formaatides transkriptsioonifailid, sealhulgas ka töös kasutatavad vtt laiendiga WebVTT failid. WebVTT failid sisaldavad ajatempleid kujul *algusaeg –> lõppaeg* ning antud ajavahemiku jooksul öeldud lausungeid. Alljärgnevalt on näide transkribeeritud faili sisust.

```
00:00:11.516 --> 00:00:13.758
oled ehk kuulnud sellist soovitust:

00:00:14.018 --> 00:00:15.939
Pead olema enesekindlam.
...
```

Transkribeerimisel ilmnes kohati mitmeid probleeme, mis nõudsid sekkumist enne tõlkimise alustamist. Süsteem, mida kasutati transkriptsioonide loomiseks, genereeris märkimisväärsel hulgal hallutsinatsioone ja korduvaid ridu, mis olid moonutatud või vigased. Mõndadel juhtudel oli transkribeeritud tekstis näha hiina tähestiku sümboleid, mis ei olnud kuidagi seotud algse heliga. Veelgi enam - mõnes failis leidus hulgaliselt korduvaid ridu, kus sisuks oli ainult üks sõna või isegi üksainus täht. Näiteks venekeelsete failide transkribeerimisel esines ühes failis 300 rida kirillitsa tähte "v". Üks näide ingliskeelses valideerimisandmestikus sisalduvate helifailide transkribeerimise tulemusel tekkinud hallutsinatsioonidest ja kordustest on toodud Lisas 4.

Nende probleemide lahendamiseks oli vaja luua skriptid, mis suudaksid tuvastada vigu transkribeeritud tekstides ja neid efektiivselt eemaldada. Skriptide loomisel tuli arvestada kõnega seonduvate asjaoludega. Loomulikus kõnes võib esineda osasid väljendeid ja sõnu, näiteks "Hästi"või "Okei", mitu korda järjest, ilma et tegu oleks hallutsinatsiooniga. Lisaks võib kõnes kohati esineda identseid paarisõnalisi lauseid mitu korda. Seetõttu tuli tekste töödeldes mitmed korduste juhud ka käsitsi üle kontrollida. Kordusi sisaldavast failist otsiti õige segment üles ning kuulati helifailis vastavat ajavahemikku. Seejärel sai langetada otsuse, kas kordusi sisaldavad read tuleks koos ajatemplitega eemaldada või andmetesse sisse jätta.

Lisaks skriptide arendamisele oli oluline ka nende testimine ja optimeerimine, et tagada nende efektiivne toimimine suurtes andmemahtudes ja erinevates olukordades. Kuna skripte tuli kasutada suurte olemasolevate andmestike peal, mille töötlemine on ressursimahukas, testiti nende töötamist mitmeid kordi erinevate probleemsete failide peal üle enne tervel andmestikul kasutamist.

Failide töötlemine

Transkribeeritud faile tuli töödelda, et viia andmestik tõlkimiseks sobivale kujule. Tõlkemootorid on hinnastatud tookeni- või tähemärgipõhiselt, seepärast on oluline anda sisendiks vaid tõlkimist vajavad tekstid. Näiteks Google Translate API-t kasutades makstakse pärin-

gus iga tähemärgi eest ja programmiliidesesse ajatemplite saatmine oleks teinud tõlkimise protsessi veel kallimaks. Transkribeeritud faile töödeldi enne tõlkimist reakaupa nii, et ajamärgised kirjutati kindlal kujul kokku ühte yaml formaadis väljundfaili. Tõlkimisse minevatest tekstifailidest eemaldati ajatemplid ning hiljem ühendati kõik ajatemplideta tekstifailid üheks failiks, et seda treenimisel kasutada.

Ajamärgiste infot sisaldav väljundfail koosneb kolmest osast: heliklipi kestus, nihe ja faili asukoht, kust igale ajamärgisele vastav kõne pärit on. Kõikide failide töötlemise jooksul luuakse mitu treenimiseks vajalikku faili: lisaks ajatemplite informatsiooni sisaldavale failile kirjutatakse kokku keeltele vastavad treenimisfailid, mis sisaldavad endas kõikide failide tekstilist sisu.

8.3.2 Kaskaadmudelid

Kaskaadmudelid, mida võrreldi, koosnesid automaatse kõnetuvastuse ja tõlkemudelite süsteemidest. Kõigepealt kasutati klastris Whisper mudeleid heliandmete transkribeerimiseks. Eesti-inglise ja eesti-vene suunal kasutati Whisper-medium-et-orthographic mudelit, inglise-eesti ja vene-eesti suunal aga Whisper-large-v3 mudelit. Seejärel rakendati väljundkaustadel terveid kaustasid töötlevaid skripte. Selleks oli loodud ülemskript, mis omakorda kutsus välja sobivat tõlkemootorit vastavalt käsureal antud argumendile ja käsureal antud tõlkesuundadele. GPT mudelite puhul oli vajalik ka mudelitele antava käsu peenhäälestamine eeldustele vastavate tõlgete saamiseks.

8.3.3 Otsemudelid

Otsemeetod-mudelid, milleks olid Whisper erinevad versioonid, SeamlessM4T ja OWSM3.1, peenhäälestati HPC-s. Kõigi mudelite peenhäälestamiseks kasutati nelja Nvidia A100 (80GB) GPUd. Peenhäälestamiseks kasutati eelnevalt loodud teksti- ja ajavahemike faile, helid segmenteeriti paarikümne sekundi pikkusteks segmentideks. Mudelite peenhäälestamiseks ja hindamiseks olid kasutusel Shell ja YAML skriptid. Alljärgnevalt on toodud näited iga mudelitüübi peenhäälestamise kohta.

Whisperi ja OWSMi puhul koosnes mudeli peenhäälestamise treeningandmestik maksimaalselt 30-sekundilistest segmentidest. Segmendid saadi ühendades mitmete järjestikuste lausungite transkriptsioonid ja lausungitele vastavad helid. SeamlessM4T mudel peenhäälestati originaalsete lausungite ja/või subtiitrite segmentidega.

Kuigi Whisper on algselt treenitud teostama ainult mitmekeelset kõnetuvastust ja kõne-

tõlget inglise keelde, suudab see tegelikkuses tõlkida kõnet ka teistesse suundadesse, kui muudetakse dekoodri prefiksit. Näiteks näitasid Peng et al. [84], et ainult käsu (*prompt*) kohandamisega võib Whisper saavutada 18.1 BLEU skoori MuST-C korpusest pärit saksa-inglise kõnetõlke valideerimisandmetel [7].

Whisperi käsu disain ei toeta alternatiivsete tõlkesuundade määratlemist. Whisper peenhäälestati kasutades "transcribe"käsku koos täiendavate kõnetõlke andmetega. Käsus täpsustatud keel vastas soovitud sihtkeelele. Järeldusfaasis mainiti käsus eeldatavat sihtkeelt, kuid lähteteksti keelt mudeli jaoks käsu sees ei täpsustatud.

Whisperit peenhäälestati kõikidel andmestikel kolme epohhi vältel. Õppimiskiirusel (*learning rate*), mille tippkiirus oli 1e-04, oli 500 soojendussammu ning pärast soojendussamme vähenes see lineaarselt nullini. Kasutusel oli plokktreening (*batch size*) 64. Stohhastilist kaalude keskmistamist (*SWA ehk Stochastic weight averaging*) [85] õppimiskiirusega 1e-05 rakendati viimasel epohhil. Lisaks kasutati Adam optimeerijat.

OWSM 3.1 EBF mudelit peenhäälestati viie epohhi jooksul, plokktreeningu suurus oli 320 ning maksimaalne õppimiskiirus oli 2.0e-04. Soojendusfaas koosnes 600 sammust. Märgendi silumise tehnikat kasutati silumisteguriga 0.1. Treeningu ajal kasutati multitegum kooder-dekooder/CTC (*Connectionist Temporal Classification*) kao meetodit. Enamik hüperparameetritest võeti ilma täiendava kohandamisteta otse ESPneti treenimisjuhistest OWSM 3.1 EBF mudeli jaoks.

SeamlessM4T peenhäälestati plokktreeningu suurusega 48. Maksimaalse õppimiskiirus oli 1e-06 koos 100 soojendussammuga. See peenhäälestamise seadistus hõlmas automaatset varast peatumist, mis mõõtis mudeli kadu väljavõetud (*heldout*) treeningandmetel pärast igat tuhandet mudeli värskendust. Treenimispotsess peatus, kui kadu ei paranenud viimase kümne hindamise jooksul. Protsessi peatumine toimus tavaliselt teise epohhi ajal.

9. Tulemused

Töö tulemusel valmis ülevaade erinevat tüüpi ja erineva ülesehitusega eeltreenitud ja peenhäälestatud mudelitest. Tulemusi võrreldakse kasutades BLEU ja BLEURT mõõdikuid. Tulemuste võrdlemiseks kasutatav BLEU skoori arvutamise meetod ei võta arvesse kirjavahemärke ja meetod vaatleb 1-4 vahemikus n-gramme. BLEU mõõdik on loomuliku keele töötlemise valdkonnas laialt kasutusel, kuid see pole võimeline tuvastama näiteks sünonüüme, ülekantud tähendustega fraase ja muid selliseid keelelisi nüansse. BLEURT mõõdik seevastu otseselt n-grammidest ei sõltu, vaid kasutab tõlke hindamiseks eeltreenitud mudelit. Kuna BLEURT võtab paremini arvesse, et ühel lausel võib olla mitu head tõlget, siis annab see lisaks BLEU mõõdikule tulemuste kohta hea ülevaate.

Tabelis 7 on toodud kõikide mudelite tulemused valideerimisandmestikel erinevatel tõlkesuundadel BLEU ja BLEURT mõõdikutega. Tabelis tähistavad "veeb" ja "sünt."erinevaid andmestikke, mida mudelite peenhäälestamiseks kasutati. "Veeb" tähistab veebist kogutud andmetest kokku pandud andmestikku, "sünt."tähistab sünteetiliselt loodud andmestikku. Tabeli esimene osa kujutab tekstist-teksti tõlkemootorite võrdlust. Tabeli teine osa kirjeldab kaskaadmudelite tulemusi. Kaskaadtõlkesüsteemide jaotise all on mudeli Whisper-medium-et-orthographic nimi tähistatud lühendatult "whisper-meo", seda kasutati automaatse kõnetuvastuse komponendina eesti-inglise ja eesti-vene suundadel. Inglise-eesti ja eesti-vene suundadel kasutati ASR komponendina Whisper-large-v3 mudelit. Tabeli kolmas osa kajastab peenhäälestamata otsemudelite tulemusi ja neljas jaotis näitab peenhäälestatud otsemudelite tulemusi.

Tabel 7. Mudelite tulemused.

Mudel	Peenhäälestatud		BLEU					BLEURT				
	veeb	sünt.	Eesti keelest		Eesti keelde			Eesti keelest		Eesti keelde		
			Inglise	Vene	Inglise	Vene	avg	Inglise	Vene	Inglise	Vene	avg
Tekstist-teksti tõlge referents-transkriptsioonidel												
Ref.-transkriptsioonid + GPT3.5-turbo	-	-	36.1	28.3	21.3	23.8	27.4	.696	.703	.593	.665	.664
Ref.-transkriptsioonid + GPT3.5-turbo-instruct	-	-	35.1	28.2	21.0	23.5	27.0	.697	.711	.702	.655	.691
Ref.-transkriptsioonid + GPT4	-	-	38.3	31.3	19.9	24.6	28.5	.702	.721	.609	.656	.672
Ref.-transkriptsioonid + Google Translate API	-	-	38.9	26.1	25.4	24.2	28.7	.690	.686	.576	.655	.652
Ref.-transkriptsioonid + NLLB-200 3.3B*	-	-	31.4	25.2	21.5	19.2	24.3	.652	.665	.529	.574	.605
Ref.-transkriptsioonid + Neurotõlge	-	-	34.8	29.3	24.7	23.7	28.1	.656	.672	.558	.619	.626
Ref.-transkriptsioonid + DeepL	-	-	34.8	25.5	29.0	25.8	28.8	.678	.695	.605	.638	.654
Kaskaad-kõnetõlkesüsteemid												
Whisper-meo/Whisper-large-v3 + GPT3.5-turbo	-	-	32.9	26.5	15.1	18.3	23.2	.649	.656	.470	.621	.599
Whisper-meo/Whisper-large-v3 + GPT3.5-turbo-instruct	-	-	32.2	25.7	15.2	18.3	22.8	.645	.649	.454	.618	.592
Whisper-meo/Whisper-large-v3 + GPT4	-	-	35.1	29.8	16.3	18.3	24.9	.647	.687	.507	.625	.617
Whisper-meo/Whisper-large-v3 + Neurotõlge	-	-	31.9	26.6	16.1	16.0	22.7	.598	.612	.458	.566	.559
Whisper-meo/Whisper-large-v3 + Google Translate API	-	-	35.2	23.8	17.4	16.1	22.9	.628	.617	.481	.585	.578
Whisper-meo/Whisper-large-v3 + NLLB-200 3.3B	-	-	28.8	23.1	15.4	13.2	20.1	.568	.568	.439	.537	.528
Peenhäälestamata otsemeetod-kõnetõlkesüsteemid												
Whisper-large-v2	-	-	17.6	-	-	-	-	.469	-	-	-	-
Whisper-large-v3	-	-	14.9	-	-	-	-	.451	-	-	-	-
SeamlessM4T v2 (large)*	-	-	13.2	16.2	6.4	13.9	12.4	.348	.426	.227	.448	.362
OWSM 3.1 EBF	-	-	0.5	0.0	1.6	0.0	0.5	.176	.153	.147	.095	.143
Peenhäälestatud otsemeetod-kõnetõlkesüsteemid												
SeamlessM4T v2 (large)	✓	-	19.3	14.4	6.1	4.3	11.0	.468	.488	.234	.261	.363
SeamlessM4T v2 (large)	-	✓	35.4	26.8	18.8	16.4	24.4	.618	.603	.482	.494	.549
SeamlessM4T v2 (large)	✓	✓	34.7	25.9	19.1	12.9	23.1	.617	.605	.470	.426	.529
Whisper-large-v3	✓	-	17.9	11.7	13.1	14.3	14.2	.496	.413	.433	.523	.466
Whisper-large-v3	-	✓	33.2	26.1	14.5	14.8	22.2	.611	.605	.363	.500	.520
Whisper-large-v3	✓	✓	33.0	25.5	17.3	16.3	23.1	.614	.603	.458	.549	.560
OWSM 3.1 EBF	-	✓	25.8	18.7	11.9	8.5	16.2	.541	.463	.377	.360	.435

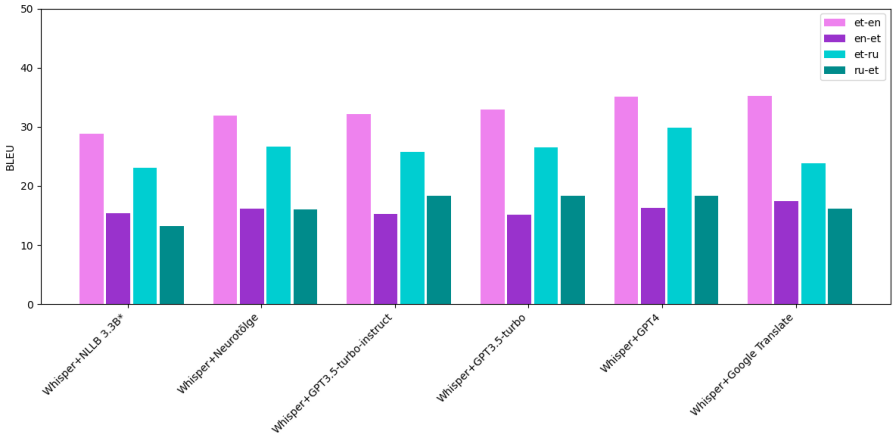
9.1 Kaskaadmudelid

Tabelis 8 on kajastatud erinevate kaskaadmudelite BLEU ja BLEURT skoorid. Värviga on märgitud parimad tulemused igal tõlkesuunal. Kaskaadmudelite tõlgete võrdluse näide on toodud Lisas 2, see koosneb ühest eesti-inglise suuna valideerimisandmestiku faili transkriptsiooni väljavõttest, sellele vastavast referentstõlkest ning kõigi kaskaadmudelite tõlgetest vastaval tekstil.

Tabelist 8 selgub, et BLEU mõõdikuga tulemuste võrdluses on kaks mudelit erinevatel tõlkesuundadel läbivalt kõige paremad. Whisper + Google Translate mudel saavutas parimad tulemused eesti-inglise ja inglise-eesti tõlkesuundadel. Whisper + GPT4 mudel saavutas parimad tulemused eesti-vene ja vene-eesti tõlkesuundadel. Eelmainitud mudel oli väga hea ka eesti ja inglise keelepaari tõlkesuundadel, näiteks eesti-inglise tõlke tulemus erines tulba parimast tulemusest vaid 0.1 punkti võrra. BLEURT mõõdiku järgi oli parimaks kaskaadmudeliks kolmel tõlkesuunal Whisper + GPT4, ainult eesti-inglise suunal oli see mudel 0.002 võrra halvem tulba parimast. BLEU ja BLEURT skooride keskmised tulemused tõlkesuundade üleselt kinnitavad samuti seda, et Whisper + GPT4 on kõige

Tabel 8. Kaskaadmudelite tulemused.

Mudel	BLEU					BLEURT				
	Eesti keelest		Eesti keelde			Eesti keelest		Eesti keelde		
	Inglise	Vene	Inglise	Vene	avg	Inglise	Vene	Inglise	Vene	avg
Whisper-meo/Whisper-large-v3 + GPT3.5-turbo	32.9	26.5	15.1	18.3	23.2	.649	.656	.470	.621	.599
Whisper-meo/Whisper-large-v3 + GPT3.5-turbo-instruct	32.2	25.7	15.2	18.3	22.8	.645	.649	.454	.618	.592
Whisper-meo/Whisper-large-v3 + GPT4	35.1	29.8	16.3	18.3	24.9	.647	.687	.507	.625	.617
Whisper-meo/Whisper-large-v3 + Neurotõlge	31.9	26.6	16.1	16.0	22.7	.598	.612	.458	.566	.559
Whisper-meo/Whisper-large-v3 + Google Translate API	35.2	23.8	17.4	16.1	22.9	.628	.617	.481	.585	.578
Whisper-meo/Whisper-large-v3 + NLLB-200 3.3B	28.8	23.1	15.4	13.2	20.1	.568	.568	.439	.537	.528



Joonis 22. Kaskaadmudelite tulemuste võrdlus.

paremate tulemustega kaskaadmudel.

Jooniselt 22 on näha, et mudelid saavad üldiselt paremini hakkama eesti keelest teise keelde tõlkimisega. Vastupidisel suunal ehk teisest keelest eesti keelde tõlkimine on mudelite jaoks keerulisem. Seda võib osaliselt põhjustada ka eesti keele käänete rohkus - kui mudel käänab tõlgitud sõna valesti, loeb BLEU selle automaatselt valeks.

9.2 Otsemudelid

Otsemeetodil põhinevatel mudelitel teostati võrdlused peenhäälestamata ja peenhäälestatud variantidel. Saadud tulemused on nähtavad Tabelis 9. Tabeli veerud kirjeldavad mudeleid, andmestikke, millel mudelit peenhäälestati, ja mõõdikuid erinevatel tõlkesuundadel. Linnukese olemasolu näitab, et seda konkreetset andmestikku kasutati mudeli

peenhäälestamiseks. Üks näide otsemudelite tõlgete võrdluse kohta eesti-inglise suuna valideerimisandmestiku faili transkriptsiooni väljavõttel on toodud Lisas 3.

Tabel 9. Otsemudelite tulemused.

Mudel	Peenhäälestatud		BLEU					BLEURT				
	veeb	sünt.	Eesti keelest		Eesti keelde			Eesti keelest		Eesti keelde		
			Inglise	Vene	Inglise	Vene	avg	Inglise	Vene	Inglise	Vene	avg
Whisper-large-v2	-	-	17.6	-	-	-	-	.469	-	-	-	-
Whisper-large-v3	-	-	14.9	-	-	-	-	.451	-	-	-	-
SeamlessM4T v2 (large)*	-	-	13.2	16.2	6.4	13.9	12.4	.348	.426	.227	.448	.362
OWSM 3.1 EBF	-	-	0.5	0.0	1.6	0.0	0.5	.176	.153	.147	.095	.143
SeamlessM4T v2 (large)	✓	-	19.3	14.4	6.1	4.3	11.0	.468	.488	.234	.261	.363
SeamlessM4T v2 (large)	-	✓	35.4	26.8	18.8	16.4	24.4	.618	.603	.482	.494	.549
SeamlessM4T v2 (large)	✓	✓	34.7	25.9	19.1	12.9	23.1	.617	.605	.470	.426	.529
Whisper-large-v3	✓	-	17.9	11.7	13.1	14.3	14.2	.496	.413	.433	.523	.466
Whisper-large-v3	-	✓	33.2	26.1	14.5	14.8	22.2	.611	.605	.363	.500	.520
Whisper-large-v3	✓	✓	33.0	25.5	17.3	16.3	23.1	.614	.603	.458	.549	.560
OWSM 3.1 EBF	-	✓	25.8	18.7	11.9	8.5	16.2	.541	.463	.377	.360	.435

9.2.1 Peenhäälestamata mudelid

Peenhäälestamata mudelid, mida töö käigus valideerimisandmestikel hinnati, olid Whisper-large v2, Whisper-large v3, SeamlessM4T v2 (large) ja OWSM 3.1 EBF. Kõikide otsemeetod-mudelite tulemustest on loodud ülevaade Tabelis 9. Tabeli esimene osa kirjeldab peenhäälestamata mudelite tulemusi. Oranžiga on tähistatud parimad peenhäälestamata mudelite tulemused. Peenhäälestuseta Whisper toimib ainult lähtekeel-inglise keel suunal ja kuna peenhäälestamiseta ei toeta see mudel teisi tõlkesuundi, on tabelis need ruudud tühjad.

Tabelist 9 on näha, et parima tulemuse eesti-inglise tõlkesuunal saavutas BLEU skoori põhjal Whisper-large-v2 mudel. Ka BLEURT mõõdiku põhjal oli antud mudel sellel suunal kõige võimekam. Suundadel eesti-vene, inglise-eesti ja vene-eesti on omavahel võrdluses vaid kaks mudelit. Kõigi kolme nimetatud suuna jaoks andis SeamlessM4T v2 (large) mudel palju paremad tulemused võrreldes OWSM 3.1 EBF mudeliga. OWSM 3.1 EBF mudel saavutas töö kõige kehvemad tulemused kõigil käsitletavatel tõlkesuundadel. BLEU skooride vahe SeamlessM4T v2 (large) ja OWSM 3.1 EBF mudeli vahel on mitmekordne. BLEURT skooride erinevus üldiselt nii suur ei ole, aga läbivalt on siiski näha, et SeamlessM4T tulemused on mitu korda paremad.

9.2.2 Peenhäälestatud otsemudelid

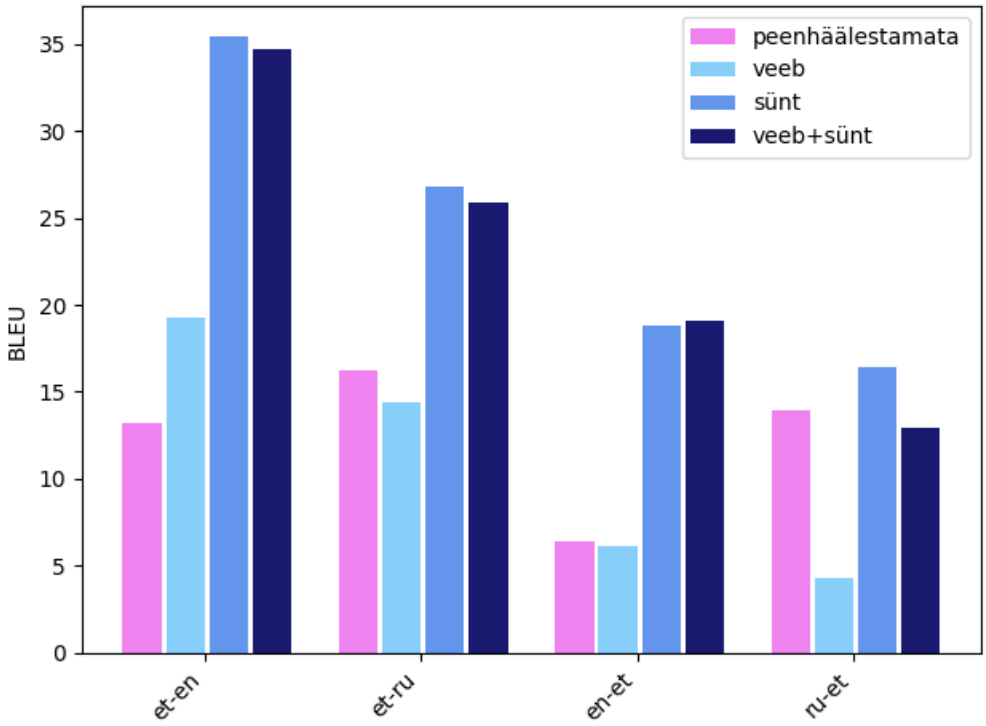
Kolme mudelit, SeamlessM4T v2, Whisper-large-v3 ja OWSM 3.1 EBF, peenhäälestati erinevatel andmestikel. Tabelis 9 on peenhäälestatud mudelite parimad tulemused märgitud heleroheliseg, linnukesed tähistavad, millise andmestikuga mudelit peenhäälestati. Peenhäälestuseks oli kolm varianti: ainult veebiandmestik, ainult sünteetiline andmestik ja nende kahe kombineerimine.

Tabelist 9 nähtub, et parim keskmine BLEU skoor on SeamlessM4T v2 (large) mudelil, mida on peenhäälestatud sünteetilistel andmetel. Mudel saavutas kõikidel suundadel parimad tulemused, välja arvatud inglise-eesti suunal, kus mudel jäi 0.3 punkti võrra tulba parimale alla. BLEURT mõõdiku keskmise tulemuse järgi on parim mudel Whisper-large-v3, mida on peenhäälestatud mõlemal andmestikul. BLEURTi puhul on näha, et SeamlessM4T (peenhäälestatud sünteetilistel andmetel) ja Whisper-large-v3 (peenhäälestatud kombineeritud andmetel) on võrdlemisi sarnaste tulemustega. Whisperi kõrgemale keskmisele aitab kaasa parem tulemus vene-eesti tõlkesuunal, kus Seamlessi tulemus on Whisperi omast 0.055 võrra väiksem. Kuna OWSM 3.1 EBF tulemused jäid ka pärast peenhäälestamist teiste mudelite tulemustele märgatavalt alla, ei jätkatud sellega teisi eksperimente.

9.2.3 Peenhäälestamise mõju

Peenhäälestamisel oli märkimisväärne mõju mudelite tõlgete kvaliteedile. Tabel 9 näitlikustas, et peenhäälestuse tulemusel võivad mudelite tulemused paraneda ligikaudu 10-25 BLEU punkti võrra. Kõige rohkem mõjutas peenhäälestamine OWSM mudelit, mis peenhäälestuseta väljastas põhimõtteliselt sisutuid tekste. Peale peenhäälestust tõusis näiteks mudeli eesti-inglise suuna tulemus 0.5-lt 25.8ni.

Peenhäälestamata ja erinevatel andmestikel peenhäälestatud SeamlessM4T v2 (large) mudeli võrdlus on toodud Joonisel 23, kus on näha, et peaaegu kõikidel tõlkesuundadel aitas mudeli tulemust kõige rohkem parandada sünteetiline andmestik. Kombineeritud andmestiku peenhäälestus andis ainsana inglise-eesti suunal veidi parema tulemuse võrreldes sünteetilise andmestikuga peenhäälestades. Üldiselt mõjus veebist kogutud andmestiku kasutamine konkreetsele mudelile siiski pigem negatiivselt. Eelnev asjaolu võib olla põhjustatud sellest, et subtiitrite ajalised piirid ei pruugi olla täpselt joondatud lausungi ajaga.



Joonis 23. Peenhäälestamata ja peenhäälestatud SeamlessM4T v2 (large) mudeli tulemused erinevatel tõlkesuundadel.

9.3 Statistiline olulisus

Tabelid 8 ja 9 näitasid, et kaskaad ja otsast lõpuni mudelite tulemuste vahed on eesti-x ja x-eesti suundade jaoks suhteliselt väikesed. Teadmata milline süsteem tõlke genereeris, võib kasutajal olla keeruline eristada, kas tõlge on loodud kaskaad- või otsesüsteemi poolt. See muudab süsteemide vahelise võrdluse ja põhjapanevate järelduste tegemise keeruliseks. Selleks, et osade mudelite tulemusi omavahel paremini võrrelda, arvutati nende BLEU skooride vahel statistiline olulisus.

Selle leidmiseks kasutati Wilcoxoni testi, et hinnata, kas skooride erinevused on statistiliselt olulised. Selleks, et näha, kas kahe süsteemi tõlgete kvaliteedi vahel on statistiliselt oluline erinevus, koguti kokku mudelite BLEU skoorid iga faili kohta valideerimisandmestikes ja arvutati iga tulemuste paari vaheline p-väärtus. Kui p-väärtus jääb alla 0.05, tähendab see seda, et kahe mudeli tulemuste vahel on statistiliselt oluline erinevus.

Tabelis 10 on kujutatud kolme väljavalitud mudeli vahelist statistilist olulisust erinevate

tõlkesuundade jaoks. Võrreldakse kaskaadsüsteemi, mis hõlmab Whisperit ja Google Translate tõlkemootorit, sünteetilistel andmetel peenhäälestatud otsast lõpuni mudelitega, milleks on Whisper-large-v3 ja SeamlessM4T. Tabelis hõlmab Whisper + Google Translate nimetuses "Whisper"kahte ASR-mudelit: Whisper-medium-et-orthographic (eesti-inglise ja eesti-vene suunal) ja Whisper-large-v3 (inglise-eesti ja vene-eesti suunal). Otsemudelite peenhäälestamist märgib lühend "ph". Mudelid, mida omavahel võrrelda, otsustati selle põhjal, et need demonstreerisid stabiilselt häid või kohati parimaid tulemusi.

Tabel 10. Statistiliselt olulised erinevused süsteemide vahel BLEU skooride põhjal.

Mudel	**Whisper + Google Translate**				**Whisper-large-v3 ph**				**SeamlessM4T ph**			
	et-en	et-ru	en-et	ru-et	et-en	et-ru	en-et	ru-et	et-en	et-ru	en-et	ru-et
Whisper + Google Translate	-	-	-	-								
Whisper-large-v3 ph					-	-	-	-				
SeamlessM4T ph									-	-	-	-

Tabelis on iga võrreldav mudel tähistatud eraldi värviga. Kõikide mudelite vahelisi seoseid hinnati kõigil neljal tõlkesuunal. Kui kahe mudeli tulemuste vahel esineb statistiline olulisus, värviti lahter selle mudeli värvi, mille tulemused on oluliselt paremad. Tulemustevahelise statistilise olulisuse puudumisel ehk juhul kui p-väärtus oli üle 0.05, on mudelite ristumise kohtades lahtrid jäetud värvimata.

Tabelist 10 on näha, et eesti-inglise tõlkesuunal esines statistiline olulisus kaskaadmudeli ja peenhäälestatud Whisper-large-v3 mudeli vahel, kaskaadmudeli tulemused olid paremad. Peenhäälestatud SeamlessM4T ja kaskaadmudeli vahel sellel tõlkesuunal statistilist olulisust ei olnud. Eesti-vene suunal näitavad tulemused, et mõlemad otsemudelid ületasid kaskaadmudeli tulemusi. Inglise-eesti ja vene-eesti tõlkesuundadel statistilist olulisust kolme mudeli tulemuste vahel ei esinenud. Tabelist on näha, et peenhäälestatud SeamlessM4T mudeli tulemused olid kõige sagedamini statistiliselt olulisemad, ületades nii kaskaadsüsteemi kui ka peenhäälestatud Whisperi tulemused eesti-vene tõlkesuunal.

10. Järeldused

Töö tulemusel loodi ülevaade eesti keelt sisaldavatest andmestikest ja pandi kokku andmestik sünteetilistest ja veebist kogutud andmetest. Seejärel teostati tõlkemootorite võrdlus, treeniti mitmel erineval meetodil põhinevad mudelid nelja tõlkesuuna jaoks ja valideeriti saadud andmeid erinevate mõõdikutega.

10.1 Eesti keele andmestikud

Uurimusest ilmnes, et eesti keele kõnetõlke jaoks sobivaid valmiskujul andmestikke on vähe. Andmestikud, mis eesti keelt mingilgi kujul sisaldavad, on tihti piiratud teemadega, loetud stiilis ja lühikese kestusega. Mõnevõrra rohkem on andmeid näiteks tekstist-teksti tõlke jaoks, kuid kuna kõnetõlke jaoks on vaja märgendatud heliandmeid, siis on sellist tüüpi andmeid vähe saada - nende loomine on mahukas ja ajanõudlik.

Internetis on suhteliselt vähe kõrge kvaliteediga kõneandmeid, millel oleksid olemas transkriptsioonid või tõlked. Mitmed allikad, mida algselt veebist kogutud lisaandmestiku jaoks koguti, tuli lõpuks välja jätta, sest kvaliteet oli halb ja mõjus mudelite peenhäälestamisele negatiivselt. Peenhäälestatud mudelite tulemuste Tabelis 9 on näha, et mõndade mudelite jõudlus väheneb peale veebist kogutud andmete juurde lisamist sünteetilistele andmetele. Kuigi veebiandmestikku kogudes lähtuti mitmetest kriteeriumitest ja vaadati mõndadel juhtudel käsitsi üle heli vastavused subtiitritega, oli üldine kvaliteet nendel siiski halvem kui sünteetilistel andmetel.

Töö jooksul kaaluti lisaandmeteks ka salvestusi erinevatelt konverentsidelt, kus reaalajas tõlgitakse esitatavaid kõnesid, et neid kuulajatele edastada. Üks selline oli näiteks TEDx Tallinn. Eelmainitud konverentsi korraldajad vastasid päringule, et nad kahjuks ürituste tõlkeid ja salvestusi ise alles ei hoia suure andmemahu tõttu. Sellistelt üritustelt oleks võimalik kõrge kvaliteediga eesti keele kõneandmeid saada.

10.2 Andmestike ja andmete iseloomu mõju mudelitele

Töö käigus loodud andmestikud mõjutasid peenhäälestatud otsemudelite tulemusi erineval määral. Nagu otsemudelite Tabelist 9 nähtus, parandas veebiandmestik peenhäälestamata mudelite tulemusi vaid mõne BLEU punkti võrra. Sünteetilise andmestikuga peenhäälestades olid tulemused märgatavalt paremad - võrreldes peenhäälestamata mudelitega

suurenesid mudelite skoorid ligikaudu 20 punkti või rohkema võrra. Mõlema andmestikuga eraldi tehtud peenhäälestamised näitasid, et sünteetilistel andmetel oli märkimisväärselt suur ja positiivne mõju mudelite tõlgetele. Veebist kogutud andmed ei parandanud mudelite tulemusi nii palju.

Huvitav oli see, kuidas mõjutas mõlema andmestiku kombineerimine mudelite peenhäälestamise tulemusi. Sünteetilistele andmetele veebiandmestiku lisamisel oli mõju mudelite tulemustele väike. Enamikel juhtudel mõjus veebiandmestiku kasutamine isegi negatiivselt, kuid mõningatel harvadel juhtudel võisid tulemused paraneda mõne punkti võrra. Eesti-inglise ja eesti-vene tõlkesuundade puhul edestas ainult sünteetilise andmestiku peenhäälestus veebiandmeid suure marginaaliga. See tuleneb osaliselt ilmselt ka sellest, et töös kasutatud eestikeelsed kõnetuvastusandmed pärinevad valideerimisandmetega sarnastest valdkondadest.

Tabel 11. SeamlessM4T (peenhäälestatud veeb + sünt.) BLEU tulemused eraldi failide kohta eesti-inglise valideerimisandmestikul.

Faili nimi	**BLEU**
16.12.2020_Tallinna_Linnavalitsuse_kolmapaevane_pressikonverents	44.9
aktuaalne-kaamera-ilm-1001-317793	38.1
aktuaalne-kaamera-ilm-1222-327710	50.0
kaamera-ilm-nadal-322248	41.7
ringvaade-2033	23.2
ringvaade-2071	23.9
Valitsuse_pressikonverents__15._oktoober_2020	26.8
	34.7

Tulemuste tõlgendamisel tuleb arvesse võtta ka valideerimisandmete iseloomu. Näiteks eesti-sihtkeel valideerimisandmestik koosneb nii uudissaadetest, mis on pigem dikteeritud iseloomuga, ja vestlussaadetest, mis on pigem poolspontaanse kõnega. Erinevat tüüpi andmetel erinesid mudelite tulemused märkimisväärselt. Üldiselt said mudelid paremini hakkama dikteeritud iseloomuga kõnega. Lisas 5 ja Tabelis 11 on näha, kui suur varieeruvus võib olla BLEU skooride ja mudeli võimekuste vahel erilaadi saadete puhul. Lisas ja tabelis toodud näide käib eesti-inglise valideerimisandmestiku failide kohta, aga mudelid käitusid sarnaselt ka teiste tõlkesuundade puhul. Jõudluse varieeruvust võib selgitada sellega, et kõne ja saated, mis suuremas osas on dikteeritud, on mudelite jaoks kergemad tõlkida. Spontaanse või poolspontaanse kõne tõlkimist on keerulisem teha, kuna sellisele kõnele

on iseloomulikud näiteks vead, mõttepausid ja juhuslik stiil.

10.3 Baasmudelid ja loodud mudelid

Käesoleva töö kontekstis on baasmudelid mudelid, mis olid juba eelnevalt olemas ja kasutajatele saadaval ehk erinevate transkribeerimissüsteemide kombineerimine tõlkemootoritega ja peenhäälestamata Whisper, SeamlessM4T ja OWSM mudelid. Nagu on Tabelis 9 näha, siis toimisid kaskaadmudelid hästi. Halvemini said hakkama otsemeetod-mudelid, mida polnud peenhäälestatud.

Töö käigus loodud peenhäälestatud otsast lõpuni arhitektuuriga mudelid saavutasid kaskaadmudelitega sarnased tulemused. Võrreldes tööeelse olukorraga on nüüd olemas käsitletud suundadele otsemudelid, mis ei kasuta tasulisi tõlkemootoreid, et saada tõlkekvaliteedilt sarnaseid tulemusi. Töö tulemused näitlikustasid, et eeltreenitud otsemudeleid on võimalik sünteesitud andmetega peenhäälestada sama heaks kui kaskaadsüsteem, aga ainult juhul, kui mudeli esialgne eesti keele kvaliteet on piisavalt hea. Lisaks selgus, et OpenAI Whisper, mis on esialgselt treenitud tegema ainult kõnetuvastust ja tõlget inglise keelde, on võimalik edukalt treenida tõlkima teiste keelepaaride vahel.

Mudelite tulemustest ilmnes, et vahed kaskaad- ja otsemeetod-süsteemide vahel on marginaalsed ning statistilist olulisust ei ole. Erinevate arhitektuuridega mudeleid võrreldi omavahel leides mudelite vahelised statistilised olulisused Tabelis 10. Tabelis on näha, et parima üldise jõudluse saavutab peenhäälestatud SeamlessM4T mudel, kuna ükski teine mudel pole üheski suunas oluliselt parem, samas edestab see nii kaskaadsüsteemi kui ka peenhäälestatud Whisperit eesti-vene tõlkesuunal.

Üldiselt võib töö tulemusel väita, et kaskaad- ja otsemeetod-süsteemid on nüüdseks eesti keele kontekstis võrdsel tasemel. Olenevalt suunast võivad vahed olla mõnest komakohast kuni mõne punkti suurused. Kui BLEU mõõdik tõi välja suuremad vahed (näiteks eesti-vene suunal 29.8 ja 26.8), siis BLEURTi mõõdiku järgi olid mõlemad tüüpi süsteemid eesti-x suundadel 0.6 läheduses. Tööst ilmnenud süsteemide sarnasus sobitub ka loomuliku kõne ja keele töötlemise valdkonna üldise suunaga, kus üha rohkem demonstreeritakse otsemeetod-mudelite võimekust ja nende saavutatud tulemuste sarnasust kaskaadmudelitega.

10.4 Eesti keelest sihtkeelde ja lähtekeelest eesti keelde

Üks töö eesmärkidest ja uurimisküsimustest oli välja selgitada, et kui hästi tõlgivad olemasolevad eeltreenitud kõnetõlkemudelid eesti keelt. Nagu ilmnes peatükis 9.2, siis eeltreenitud mudelitest suutis ainsana kõigil tõlkesuundadel arvestatava tulemusega tõlkeid väljastada SeamlessM4T v2 large mudel. Eesti-inglise suunal toimis võrdlemisi hästi ka Whisper-large-v2, kuid teistele tõlkesuundadele see mudel ei kohandu.

Mudelite tulemusi on kohati keeruline kõrvutada, kuna erinevate tõlkesuundade vahel ei ole mõõdikute tulemused otseselt võrreldavad. See tuleneb erinevate keelte omadustest, näiteks inglise keelel on 3 käänet ja eesti keelel hoopis 14 käänet. Kui mudel tõlgib inglise keelest eesti keelde sõna ja teeb vea käände otsustamisel, on automaatsete mõõdikute jaoks tegu vale sõnaga, kuigi sisuliselt oli tegemist peaaegu korrektse tõlkega. Samuti on antud keelepaari vahel erinevused näiteks järgmiste omaduste olemasolus: nimisõnade artiklid (the, a, an), grammatiline sugu (he, she, they), ajavormid (eesti keeles otseselt tulevikku ei ole) ja palju muud. Sarnaseid erinevusi keelte omadustes on ka teistel keelepaaridel.

Üldiselt saavutasid mudelid paremaid tulemusi, kui tõlkesuunaks oli eesti keelest sihtkeelde. Väheste ressurssidega keel on lihtsamini tõlgitav inglise ja vene keelde kui vastupidi. Otsemudelid on tavaliselt treenitud suurel hulgal kõrge ressursiga keelte (näiteks inglise- ja venekeelsetel) andmetel, mis aitab mudelitel paremaid tõlkeid luua. Kuna andmestikes pole vähese ressurssiga keeli sellises mahus, saavadki mudelid paremini hakkama, kui väljundkeel on kõrge ressursiga keel.

10.5 Edasine töö

Käesolev töö annab aluse erinevateks edasisteks katsetusteks ja uurimiseks ning pakub mitmeid suundi, millega tulevikus tööd jätkata.

Kaskaadmudeli puhul saab tulevikus eksperimenteerida erinevate transkribeerimistehnoloogiatega. Hetkel on töös kasutatud ainult Whisperi transkribeerimissüsteeme, kuid näiteks SeamlessM4T toetab ka automaatset kõnetuvastust ligi 100 keele jaoks. Kaskaadmudelite puhul saab kaaluda lisaks teistsuguste tõlkemootorite kasutamist. Mõne vähemtuntud tõlkemootori kasutamisel võib olla võimalik saavutada konkurentsivõimelise kvaliteediga tõlkeid. Selliste katsete käigus saab tuvastada parimad tõlkemootorid ning edaspidi on võimalik need kasutusele võtta kaskaadsüsteemi masintõlke komponendina või kasutada neid sünteetiliste andmete genereerimiseks. Sarnaselt on võimalik viia läbi edasiseid eksperimente ka teiste otsast lõpuni mudelitega, mida antud töös ei kasutatud.

Kui saada veelgi kvaliteetsemad andmed, oleks võimalik ka paremaid otsast lõpuni arhitektuuriga mudeleid treenida. Siiski, nagu eelnevalt on töös kirjeldatud, on sellisel kujul kõrge kvaliteediga andmeid väga vähe saadaval. Töö käigus loodud sünteetiline andmestik koosneb peamiselt täiskasvanud inimeste kõnest. Andmestikku saaks muuta mitmekesisemaks, tekitades sinna rohkem varieeruvust - koguda rohkem heliandmeid, mille kõnelejad oleksid varieeritud (vanus, sugu jms).

Loodud andmestikku on tulevikus võimalik kasutada mudelite treenimise ja peenhäälestamise protsessis, et seeläbi saavutada kõnetõlkes paremaid tulemusi. Loodud otsemudeleid on võimalik kasutada eestikeelse kõne tõlkimisel inglise või vene keelde ja vastupidi. Edaspidi oleks võimalik uurida eesti keele sünkroontõlget teostavate otsast lõpuni mudelite loomist.

11. Kokkuvõte

Kõnetõlke valdkonnas on hetkel levinud kaks peamist lähenemist: kaskaad- ja otsemeetod. Kaskaadmudelid koosnevad eraldi automaatse kõnetuvastuse süsteemist ja tõlkesüsteemist. Sisendheli transkribeeritakse teksti kujule ja seejärel tõlgib tõlkesüsteem antud lähtekeelse teksti sihtkeelseks tekstiks. Otsemeetod-mudelite puhul tõlgitakse lähtekeelne sisendkõne otse sihtkeelseks tekstiväljundiks. Nendel mudelitel pole eraldi kõne transkribeerimise etappi, vaid kõnetõlge tehakse ühtses mudelis. Töö eesmärk oli võrrelda kaskaad- ja otsemeetod-mudeleid eestikeelse suulise kõne tõlkimise kontekstis. Selleks analüüsiti olemasolevaid kõnetõlke andmestikke ja loodi ülevaade eesti keele esindatusest antud andmestikes. Töö käigus sünteesiti andmeid, tõlkides heliandmetest automaatse kõnetuvastusega loodud tekstifailid sihtkeelde, ja koguti veebist andmeid, et luua veebiandmestik. Seejärel viidi läbi katsed erinevate kaskaad- ja otsemeetod-mudelitega, otsemeetod-mudeleid hinnati nii peenhäälestamata kui ka peenhäälestatud versioonides.

Kõnetõlkeks valmiskujul saadaolevaid andmestikke, mis sisaldaksid eesti keelt, on vähe. Kui andmestik sisaldab eesti keelt, on see väheses mahus ja tihtipeale loetud stiilis. Töö tulemusel valmis andmestik, mida on võimalik edaspidi kasutada eesti-inglise ja eesti-vene keelepaaride kõne ja teksti tõlkimise süsteemide parendamiseks. Andmestik koosneb sünteetilistest andmetest ja internetist kogutud lisaandmetest. Sünteetiline andmestik on eesti-x suunal 1300h, inglise-eesti suunal umbes 1000h ja vene-eesti suunal 102h. Veebiandmestik on eesti-inglise suunal 40h, eesti-vene suunal 18h, inglise-eesti suunal 58h ja vene-eesti suunal 617h. Töö käigus läbiviidud eksperimentidest nähtus, et sünteetiliste andmete kasutamine parandab mudelite tulemusi märgatavalt.

Tõlkemootorid, mida kaskaadmudelite osadena käsitleti, olid järgnevad: GPT3.5-turbo, GPT3.5-turbo-instruct, GPT4, Neurotõlge, Google Translate ja NLLB-200 3.3B. Otsemeetod-mudelid olid järgnevad: Whisper large v2 ja v3, SeamlessM4T v2 ja OWSM 3.1. Mudelite võrdlused teostati neljal suunal: eesti-inglise, eesti-vene, inglise-eesti ja vene-eesti. Valideerimisandmestikud koosnesid peamiselt vestlussaadetest, pressikonverentsidest ja uudissaadetest. Valideerimisandmed transkribeeriti käsitsi ja transkriptsioonid tõlgiti tõlkebüroode poolt.

Tõlkesuundade vahel varieerusid erinevate mudelite tulemused vähesel määral. Nii kaskaadsüsteem, mis koosnes Whisper + Google Translate mudelitest, kui ka sünteetilistel andmetel peenhäälestatud SeamlessM4T otsemudel saavutasid valideerimisandmetel sarna-

seid tulemusi. Väikesed vahed olid tõlkesuundade vahel - näiteks inglise-eesti tõlkesuunal saavutas parema BLEU skoori otsemudel, kuid eesti-vene tõlkesuunal saavutas 3 punkti võrra parema tulemuse kaskaadsüsteem. BLEURT mõõdiku tulemusi vaadeldes muutuvad vahed süsteemide vahel veelgi väiksemaks.

Kuigi tõlkesuundade vahel olid väikesed erinevused, toimisid kaskaad- ja otsemeetod mudelid siiski võrdlemisi sarnastel tasemetel. Töö tulemusel võib järeldada, et eesti keele puhul toimivad mõlemat tüüpi mudelid heal tasemel. Kaskaadsüsteemi, peenhäälestatud Whisperit ja peenhäälestatud SeamlessM4T mudelit võrreldi omavahel, leides mudelite vahelised statistilised olulisused Wilcoxoni testiga. Võrdlus näitas, et peenhäälestatud SeamlessM4T toimib läbivalt kõige paremini, sest mudel annab eesti-vene tõlkesuunal paremaid tulemusi kui teised mudelid ning ülejäänud tõlkesuundadel statistilist olulisust ei ilmnenud.

Töö tulemusel valminud andmestik ja mudelid jagatakse avaandmete ja vabavarana. Töö tulemused on panus loomuliku kõne töötlemise valdkonna teadmisesse.

Kasutatud kirjandus

[1] World Health Organization. *World report on hearing*. Global report. Sensory Functions, Disability ja Rehabilitation (SDR), 2021, lk. 40. URL: https://www.who.int/publications/i/item/9789240020481.

[2] Matthias Sperber. „End-to-End Neural Speech Translation“. Doctoral Thesis. Karlsruhe Institute of Technology, 2019. URL: https://aclanthology.org/C88-2167.

[3] Luisa Bentivogli *et al. Cascade versus Direct Speech Translation: Do the Differences Still Make a Difference?* 2021. arXiv: 2106.01045 [cs.CL].

[4] Seamless Communication *et al. SeamlessM4T: Massively Multilingual I& Multimodal Machine Translation*. 2023. arXiv: 2308.11596 [cs.CL].

[5] Nathaniel Robinson *et al.* „ChatGPT MT: Competitive for High- (but Not Low-) Resource Languages“. Teoses: *Proceedings of the Eighth Conference on Machine Translation*. Toim. Philipp Koehn *et al.* Singapore: Association for Computational Linguistics, detsember 2023, lk. 392–418. DOI: 10.18653/v1/2023.wmt-1.40. URL: https://aclanthology.org/2023.wmt-1.40.

[6] Vassil Panayotov *et al.* „Librispeech: An ASR corpus based on public domain audio books“. Teoses: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, lk. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964.

[7] Mattia A. Di Gangi *et al.* „MuST-C: a Multilingual Speech Translation Corpus“. Teoses: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Toim. Jill Burstein, Christy Doran ja Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, juuni 2019, lk. 2012–2017. DOI: 10.18653/v1/N19-1202. URL: https://aclanthology.org/N19-1202.

[8] Hiroyuki Kaji. „An Efficient Execution Method for Rule-Based Machine Translation“. Teoses: *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*. 1988. URL: https://aclanthology.org/C88-2167.

[9] Sergei Nirenburg, Constantine Domashnev ja Dean J. Grannes. „Two Approaches to Matching in Example-Based Machine Translation“. Teoses: *Proceedings of the Fifth Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*. Kyoto, Japan, 1993. URL: https://aclanthology.org/1993.tmi-1.4.

[10] Peter F. Brown *et al.* „The Mathematics of Statistical Machine Translation: Parameter Estimation“. *Computational Linguistics* 19.2 (1993). Toim. Julia Hirschberg, lk. 263–311. URL: https://aclanthology.org/J93-2003.

[11] Hendra Setiawan *et al.* „Phrase-Based Statistical Machine Translation: A Level of Detail Approach“. Teoses: *Second International Joint Conference on Natural Language Processing: Full Papers*. 2005. DOI: 10.1007/11562214_51. URL: https://aclanthology.org/I05-1051.

[12] Holger Schwenk, Daniel Dechelotte ja Jean-Luc Gauvain. „Continuous Space Language Models for Statistical Machine Translation“. Teoses: *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Sydney, Australia: Association for Computational Linguistics, juuli 2006, lk. 723–730. URL: https://aclanthology.org/P06-2093.

[13] Nal Kalchbrenner ja Phil Blunsom. „Recurrent Continuous Translation Models“. Teoses: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Toim. David Yarowsky *et al.* Seattle, Washington, USA: Association for Computational Linguistics, oktoober 2013, lk. 1700–1709. URL: https://aclanthology.org/D13-1176.

[14] Ilya Sutskever, Oriol Vinyals ja Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: 1409.3215 [cs.CL].

[15] Kyunghyun Cho *et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: 1406.1078 [cs.CL].

[16] Dzmitry Bahdanau, Kyunghyun Cho ja Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2015. arXiv: 1409.0473 [cs.CL].

[17] Ashish Vaswani *et al. Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[18] Mattia Di Gangi, Matteo Negri ja Marco Turchi. „Adapting Transformer to End-to-End Spoken Language Translation“. Teoses: september 2019, lk. 1133–1137. DOI: 10.21437/Interspeech.2019-3045.

[19] Anmol Gulati *et al. Conformer: Convolution-augmented Transformer for Speech Recognition*. 2020. arXiv: 2005.08100 [eess.AS].

[20] Yifan Peng *et al. Branchformer: Parallel MLP-Attention Architectures to Capture Local and Global Context for Speech Recognition and Understanding*. 2022. arXiv: 2207.02971 [cs.CL].

[21] Steffen Schneider *et al. wav2vec: Unsupervised Pre-training for Speech Recognition*. 2019. arXiv: 1904.05862 [cs.CL].

[22] Alexei Baevski *et al. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. 2020. arXiv: 2006.11477 [cs.CL].

[23] Wei-Ning Hsu *et al. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units*. 2021. arXiv: 2106.07447 [cs.CL].

[24] Chen Xu *et al. Recent Advances in Direct Speech-to-text Translation*. 2023. arXiv: 2306.11646 [cs.CL].

[25] Alexandre Berard *et al. Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation*. 2016. arXiv: 1612.01744 [cs.CL].

[26] Ron J. Weiss *et al. Sequence-to-Sequence Models Can Directly Translate Foreign Speech*. 2017. arXiv: 1703.08581 [cs.CL].

[27] Thierry Etchegoyhen *et al.* „Cascade or Direct Speech Translation? A Case Study“. *Applied Sciences* 12.3 (2022). ISSN: 2076-3417. DOI: 10.3390/app12031097. URL: https://www.mdpi.com/2076-3417/12/3/1097.

[28] Milind Agarwal *et al.* „FINDINGS OF THE IWSLT 2023 EVALUATION CAMPAIGN“. Teoses: *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*. Toim. Elizabeth Salesky, Marcello Federico ja Marine Carpuat. Toronto, Canada (in-person ja online): Association for Computational Linguistics, juuli 2023, lk. 1–61. DOI: 10.18653/v1/2023.iwslt-1.1. URL: https://aclanthology.org/2023.iwslt-1.1.

[29] Antonios Anastasopoulos *et al.* „FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN“. Teoses: *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*. Toim. Marcello Federico *et al.* Bangkok, Thailand (online): Association for Computational Linguistics, august 2021, lk. 1–29. DOI: 10.18653/v1/2021.iwslt-1.1. URL: https://aclanthology.org/2021.iwslt-1.1.

[30] Antonios Anastasopoulos *et al.* „Findings of the IWSLT 2022 Evaluation Campaign“. Teoses: *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*. Toim. Elizabeth Salesky, Marcello Federico ja Marta Costa-jussà. Dublin, Ireland (in-person ja online): Association for Computational Linguistics, mai 2022, lk. 98–157. DOI: 10.18653/v1/2022.iwslt-1.10. URL: https://aclanthology.org/2022.iwslt-1.10.

[31] Long Ouyang *et al. Training language models to follow instructions with human feedback*. 2022. arXiv: 2203.02155 [cs.CL].

[32] Tom B. Brown *et al. Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[33] Siddharth Sharma, Simone Sharma ja Anidhya Athaiya. „ACTIVATION FUNCTIONS IN NEURAL NETWORKS“. *International Journal of Engineering Applied Sciences and Technology* (2020). URL: https://api.semanticscholar.org/CorpusID:225922639.

[34] Taiwo Ayodele. „Types of Machine Learning Algorithms“. Teoses: veebruar 2010. ISBN: 978-953-307-034-6. DOI: 10.5772/9385.

[35] Murat Sazli. „A brief review of feed-forward neural networks“. *Communications Faculty Of Science University of Ankara* 50 (jaanuar 2006), lk. 11–17. DOI: 10.1501/commua1-2_0000000026.

[36] Haşim Sak, Andrew Senior ja Françoise Beaufays. *Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition*. 2014. arXiv: 1402.1128 [cs.NE].

[37] Sepp Hochreiter ja Jürgen Schmidhuber. „Long Short-term Memory“. *Neural computation* 9 (detsember 1997), lk. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[38] Junyoung Chung *et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: 1412.3555 [cs.NE].

[39] Kyunghyun Cho *et al. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. 2014. arXiv: 1409.1259 [cs.CL].

[40] Alec Radford ja Karthik Narasimhan. „Improving Language Understanding by Generative Pre-Training“. Teoses: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[41] Mike Lewis *et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL].

[42] Jacob Devlin *et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[43] Yinhan Liu *et al. Multilingual Denoising Pre-training for Neural Machine Translation*. 2020. arXiv: 2001.08210 [cs.CL].

[44] Yuqing Tang *et al. Multilingual Translation with Extensible Multilingual Pretraining and Finetuning*. 2020. arXiv: 2008.00401 [cs.CL]. URL: https://arxiv.org/abs/2008.00401.

[45] Thilo von Neumann *et al.* „On Word Error Rate Definitions and Their Efficient Computation for Multi-Speaker Speech Recognition Systems“. Teoses: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023. DOI: 10.1109/icassp49357.2023.10094784. URL: http://dx.doi.org/10.1109/ICASSP49357.2023.10094784.

[46] Kishore Papineni *et al.* „BLEU: a method for automatic evaluation of machine translation“. Teoses: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002, lk. 311–318. DOI: 10.3115/1073083.1073135. URL: https://doi.org/10.3115/1073083.1073135.

[47] Thibault Sellam, Dipanjan Das ja Ankur Parikh. „BLEURT: Learning Robust Metrics for Text Generation“. Teoses: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Toim. Dan Jurafsky *et al.* Online: Association for Computational Linguistics, juuli 2020, lk. 7881–7892. DOI: 10.18653/v1/2020.acl-main.704. URL: https://aclanthology.org/2020.acl-main.704.

[48] Frank Wilcoxon. „Individual Comparisons by Ranking Methods“. *Biometrics Bulletin* 1.6 (1945), lk. 80–83. ISSN: 00994987. URL: http://www.jstor.org/stable/3001968 (vaadatud 02.05.2024).

[49] *Wilcoxon signed-rank test*. https://datatab.net/tutorial/wilcoxon-test. [Vaadatud: 04-05-24].

[50] Alvan R. Feinstein. „P-Values and Confidence Intervals: Two Sides of the Same Unsatisfactory Coin“. *Journal of Clinical Epidemiology* 51.4 (1998), lk. 355–360. ISSN: 0895-4356. DOI: https://doi.org/10.1016/S0895-4356(97)00295-3. URL: https://www.sciencedirect.com/science/article/pii/S0895435697002953.

[51] Thomas Zenkel *et al.* „Open Source Toolkit for Speech to Text Translation“. *The Prague Bulletin of Mathematical Linguistics* 111 (oktoober 2018), lk. 125–135. DOI: 10.2478/pralin-2018-0011.

[52] Long Duong *et al.* „An Attentional Model for Speech Translation Without Transcription“. Teoses: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Toim. Kevin Knight, Ani Nenkova ja Owen Rambow. San Diego, California: Association for Computational Linguistics, juuni 2016, lk. 949–959. DOI: 10.18653/v1/N16-1109. URL: https://aclanthology.org/N16-1109.

[53] Sameer Bansal *et al. Low-Resource Speech-to-Text Translation*. 2018. arXiv: 1803.09164 [cs.CL].

[54] Matthias Sperber ja Matthias Paulik. „Speech Translation and the End-to-End Promise: Taking Stock of Where We Are“. Teoses: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Toim. Dan Jurafsky *et al.* Online: Association for Computational Linguistics, juuli 2020, lk. 7409–7421. DOI: 10.18653/v1/2020.acl-main.661. URL: https://aclanthology.org/2020.acl-main.661.

[55] Mattia Di Gangi, Matteo Negri ja Marco Turchi. „Adapting Transformer to End-to-End Spoken Language Translation“. Teoses: september 2019, lk. 1133–1137. DOI: 10.21437/Interspeech.2019-3045.

[56] Parnia Bahar, Tobias Bieschke ja Hermann Ney. *A Comparative Study on End-to-end Speech to Text Translation*. 2019. arXiv: 1911.08870 [cs.CL].

[57] Alec Radford *et al. Robust Speech Recognition via Large-Scale Weak Supervision*. 2022. arXiv: 2212.04356 [eess.AS].

[58] Yifan Peng *et al. Reproducing Whisper-Style Training Using an Open-Source Toolkit and Publicly Available Data*. 2023. arXiv: 2309.13876 [cs.CL].

[59] Yifan Peng *et al.* „OWSM v3.1: Better and Faster Open Whisper-Style Speech Models based on E-Branchformer“. *ArXiv* abs/2401.16658 (2024). URL: https://api.semanticscholar.org/CorpusID:267320798.

[60] NLLB Team *et al. No Language Left Behind: Scaling Human-Centered Machine Translation*. 2022. arXiv: 2207.04672 [cs.CL].

[61] Peter J. Liu *et al. Generating Wikipedia by Summarizing Long Sequences*. 2018. arXiv: 1801.10198 [cs.CL].

[62] Alec Radford *et al.* „Language Models are Unsupervised Multitask Learners“. Teoses: 2019. URL: https://api.semanticscholar.org/CorpusID:160025533.

[63] *Introducing ChatGPT*. https://openai.com/blog/chatgpt. [Vaadatud: 24-04-2024].

[64] *OpenAI's text generation models*. https://platform.openai.com/docs/guides/text-generation/text-generation-models. [Vaadatud: 24-04-24].

[65] OpenAI *et al. GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL].

[66] *DeepL Translator*. https://www.deepl.com/whydeepl. [Vaadatud: 30-04-24].

[67] *TartuNLP Neurotõlge*. https://neurotolge.ee. [Vaadatud: 28-04-24].

[68] *Google Translation AI*. https://cloud.google.com/translate?hl=en. [Vaadatud: 30-04-24].

[69] Rosana Ardila *et al.* „Common Voice: A Massively-Multilingual Speech Corpus“. English. Teoses: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Toim. Nicoletta Calzolari *et al.* Marseille, France: European Language Resources Association, mai 2020, lk. 4218–4222. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.520.

[70] Mozilla. *Common Voice*. [Vaadatud: 30-04-24]. 2022. URL: https://commonvoice.mozilla.org/et.

[71] Changhan Wang *et al.* „CoVoST: A Diverse Multilingual Speech-To-Text Translation Corpus“. English. Teoses: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, mai 2020, lk. 4197–4203. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.517.

[72] Changhan Wang, Anne Wu ja Juan Pino. *CoVoST 2 and Massively Multilingual Speech-to-Text Translation*. 2020. arXiv: 2007.10310 [cs.CL].

[73] Ye Jia *et al.* „CVSS Corpus and Massively Multilingual Speech-to-Speech Translation“. Teoses: *Proceedings of Language Resources and Evaluation Conference (LREC)*. 2022, lk. 6691–6703.

[74] Alexis Conneau *et al. FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech*. 2022. arXiv: 2205.12446 [cs.CL].

[75] Elizabeth Salesky *et al.* „Multilingual TEDx Corpus for Speech Recognition and Translation“. Teoses: *Proceedings of Interspeech*. 2021.

[76] Jörgen Valk ja Tanel Alumäe. „VoxLingua107: a Dataset for Spoken Language Recognition“. Teoses: *Proc. IEEE SLT Workshop*. 2021.

[77] Alan W Black. „CMU Wilderness Multilingual Speech Dataset“. Teoses: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, lk. 5971–5975. DOI: 10.1109/ICASSP.2019.8683536.

[78] Vineel Pratap *et al.* „MLS: A Large-Scale Multilingual Dataset for Speech Research“. Teoses: *Interspeech 2020*. interspeech$_2$020. ISCA, oktoober 2020. DOI: 10.21437/interspeech.2020-2826. URL: http://dx.doi.org/10.21437/Interspeech.2020-2826.

[79] Javier Iranzo-Sánchez *et al. Europarl-ST: A Multilingual Corpus For Speech Translation Of Parliamentary Debates*. 2020. arXiv: 1911.03167 [cs.CL].

[80] Changhan Wang *et al. VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation*. 2021. arXiv: 2101.00390 [cs.CL].

[81] Guoguo Chen *et al. GigaSpeech: An Evolving, Multi-domain ASR Corpus with 10,000 Hours of Transcribed Audio*. 2021. arXiv: 2106.06909 [cs.SD].

[82] Tanel Alumäe *et al.* „Automatic Closed Captioning for Estonian Live Broadcasts“. Teoses: *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*. Tórshavn, Faroe Islands: University of Tartu Library, mai 2023, lk. 492–499. URL: https://aclanthology.org/2023.nodalida-1.49.

[83] *TalTech Estonian Speech Dataset 1.0*. https://cs.taltech.ee/staff/tanel.alumae/data/est-pub-asr-data/. [Vaadatud: 04-05-24].

[84] Puyuan Peng *et al. Prompting the Hidden Talent of Web-Scale Speech Models for Zero-Shot Task Generalization*. 2023. arXiv: 2305.11095 [eess.AS].

[85] Pavel Izmailov *et al. Averaging Weights Leads to Wider Optima and Better Generalization*. 2019. arXiv: 1803.05407 [cs.LG].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Tiia Sildam, Andra Velve

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Kaskaad- ja otsemeetodi võrdlus eesti keele suulise kõne tõlkesüsteemide näitel", mille juhendaja on Tanel Alumäe
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

04.06.2024

[1] Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Kaskaadmudelite tõlgete võrdlus eesti-inglise tõlkesuunal

Allpool tabelis on toodud kaskaadmeetodil mudelite tõlgete võrdlused. Tabel sisaldab käsitsi transkribeeritud teksti saatest "Ringvaade"ning selle tõlkeid. Saate tekst pärineb eesti- inglise suuna valideerimisandmestiku failist ringvaade-2033-320571.et.

Algne tekst	Tänastes välisuudistes näitame teile muuhulgas maailma vanimat profijalgpallurit ja maja, millele pandi jalad alla. Tänaseid välisuudiseid alustame sellega, kuidas maja jalad alla võtab, sõna otseses mõttes. Sellise imeteoga saadi hakkama Hiinas Shanghais, kus seitse tuhat kuussada tonni kaaluvale viiekorruselisele koolile, sada üheksakümmend kaheksa robotjalga alla monteeriti ja jalutati temaga täpselt kuuskümmend üks koma seitse meetrit.	
Mudel	**Tõlge**	**BLEU**
Referents	In today's foreign news, we will show you, among other things, the world's oldest professional footballer and a house that was mounted on legs. We start today's foreign news with a report on a house that has, literally, stood on its own legs. Such a miracle was achieved in Shanghai, China, where a five-story school building, weighing seven thousand six hundred tons, was mounted on one hundred ninety-eight robotic legs and was 'walked' for exactly sixty-one point seven meters.	-

Mudel	Tõlge	BLEU
Whisper-meo + GPT3.5-turbo	In today's international news, we will show you, among other things, the world's oldest professional football players and a house with legs. We start today's international news with the story of a house taking legs. Quite literally, this incredible feat was accomplished in Shanghai, China, where one hundred and ninety-eight robot legs were assembled and attached to a five-story school weighing seven thousand six hundred tons. The school was then walked precisely sixty-one point seven meters.	32.9
Whisper-meo + GPT3.5-turbo-instruct	In today's international news, we will show you, among other things, the world's oldest professional football players and a house that was given legs. We start today's international news with the story of how a house gets legs. Literally, this miracle was achieved in Shanghai, China, where one hundred and ninety-eight robot legs were assembled under a five-story school weighing seven thousand six hundred tons and walked with it for exactly sixty-one point seven meters.	32.2
Whisper-meo + GPT4	In today's foreign news, we will show you, among other things, the world's oldest professional footballers and a house that was put on legs. We start today's foreign news with how a house takes on legs. Literally such a miracle was accomplished in Shanghai, China, where one hundred and ninety-eight robotic legs were mounted under a five-story school weighing seven thousand six hundred tons and it was walked exactly sixty-one point seven meters.	35.1

Mudel	Tõlge	BLEU
Whisper-meo + Neurotõlge	In today's foreign news, we will show you, among other things, the world's oldest professional footballers and the house that was put under your feet. Today's foreign news, let's start with the way the house gets its feet down. Literally such a miracle was accomplished in Shanghai, China, where a five-story school weighing seven thousand and six hundred tons was mounted under a hundred and ninety-eight robot feet and walked with him exactly sixty-one decimal seven meters.	31.9
Whisper-meo + Google Translate API	In today's foreign news, we show you, among other things, the world's oldest professional football players and the house where they set foot. We start today's foreign news with how the house takes off. Literally, such a miracle was achieved in Shanghai, China, where one hundred and ninety-eight robot legs were mounted on a five-story school weighing seven thousand six hundred tons and walked with him exactly sixty one point seven meters.	34.7
Whisper-meo + NLLB-200 3.3B	In today's outside shots, we'll show you, among other things, the world's oldest professional footballers and the house where they set foot. We're going to start today's exhibits by taking a look at how a building is being pulled down, a literally miraculous feat that was accomplished in Shanghai, China, where one hundred and ninety-eight robotic feet were installed underneath a five-story school weighing seven thousand six hundred tons. And they walked him exactly sixty-one cubic feet...	28.8

Lisa 3 – Otsemudelite tõlgete võrdlus eesti-inglise tõlkesuunal

Allpool tabelis on toodud otsemeetod-mudelite tõlgete võrdlused. Tabel sisaldab käsitsi transkribeeritud teksti saatest "Ringvaade"ning selle tõlkeid. Saate tekst pärineb eesti-inglise suuna valideerimisandmestiku failist *ringvaade-2033-320571.et*.

Algne tekst	Tänastes välisuudistes näitame teile muuhulgas maailma vanimat profijalgpallurit ja maja, millele pandi jalad alla. Tänaseid välisuudiseid alustame sellega, kuidas maja jalad alla võtab, sõna otseses mõttes. Sellise imeteoga saadi hakkama Hiinas Shanghais, kus seitse tuhat kuussada tonni kaaluvale viiekorruselisele koolile, sada üheksakümmend kaheksa robotjalga alla monteeriti ja jalutati temaga täpselt kuuskümmend üks koma seitse meetrit.	
Mudel	**Tõlge**	**BLEU**
Referents	In today's foreign news, we will show you, among other things, the world's oldest professional footballer and a house that was mounted on legs. We start today's foreign news with a report on a house that has, literally, stood on its own legs. Such a miracle was achieved in Shanghai, China, where a five-story school building, weighing seven thousand six hundred tons, was mounted on one hundred ninety-eight robotic legs and was 'walked' for exactly sixty-one point seven meters.	-
Whisper-large-v2	In today's breaking news, we will show you the world's oldest professional footballers and the house to which they put their feet. Today's breaking news. Let's start with how the house puts its feet. In short, such a masterpiece began in Shanghai, China, where a 7600-tonne house for a five-story school was installed with 198 robots under its feet and walked exactly 61.7 metres with it.	17.6

Mudel	Tõlge	BLEU
Whisper-large-v3	In today's episode, we'll show you the world's oldest professional football players and the house where they put their feet down. In today's episode, we'll start with how the house takes its feet down. In short, such a miracle happened in Shanghai, China, where 7600 tons of material was mounted on a 5-storey school and 198 robots' feet were put down and walked exactly 61.7 meters with it.	14.9
SeamlessM4T v2 (large)*	In today's foreign news, we show you, among other things, the world's oldest professional footballer and the house he walked into. It's hard to imagine what it would have been like to live in Shanghai, China, where a 7,600-square-foot, five-story school building was walked through by a robot, leaving behind a school building that would have been demolished in the 1930s.	13.2
OWSM 3.1 EBF	The first one is the only one who has the most high-quality profilal pallority on the main, and the main one is the other. If you are going to see the world, you will see the world's best, you will see the world's best, and you will see the world's best ...	0.5
SeamlessM4T v2 (large), peenhäälestatud veeb	In today's foreign news, we will show you the world's oldest professional footballers and the house they set foot in. Today's foreign news starts with how the building collapses. Such a miracle was done in Shanghai, China, where a 7600 ton five-story school was installed and walked with exactly 61.7 meters.	19.3

Mudel	Tõlge	BLEU
SeamlessM4T v2 (large), peenhäälestatud sünt.	In today's foreign news, we will show you, among other things, the world's oldest professional football player and the house on which they set foot. Today's foreign news, let's start with how the house gets its feet down. Literally, such a miracle was accomplished in Shanghai, China, where one hundred and ninety-eight robot feet were installed in a five-story school weighing seven thousand six hundred tons and walked with it for exactly sixty-one point seven meters.	35.4
SeamlessM4T v2 (large), peenhääles-tatud veeb + sünt.	In today's foreign news, we show you, among other things, the world's oldest professional footballer and the house on which the feet were put. Today's foreign news, let's start with how the house takes off. Literally, such a miracle was managed in Shanghai, China, where one hundred and ninety-eight robot legs were installed in a five-story school weighing seven thousand six hundred tons and walked with it exactly sixty-one point seven meters.	34.7
Whisper-large-v3, peenhäälestatud veeb	In today's news we will show you the world's oldest football players and a house that was put under the feet. Today's news. Let's start with how the house takes the feet down. In short, such a wonder was started in China, Shanghai, where a 5-storey school with a weight of 7600 tons had to be remonted under the feet of a robot and it was walked down to exactly 61,7 m.	17.9
Whisper-large-v3, peenhäälestatud sünt.	In today's foreign news, we show you, among other things, the world's oldest professional footballer and a house that was put on foot. Today's foreign news, let's start with how the house takes its feet, literally. Such a miracle was managed in Shanghai, China, where a hundred and ninety-eight robot feet were assembled and walked exactly sixty-one point seven meters for a five-story school weighing seven thousand six hundred tons.	33.2

Mudel	Tõlge	BLEU
Whisper-large-v3, peenhääles-tatud veeb + sünt.	In today's foreign news, we show you, among other things, the world's oldest professional football player and a house that was put under its feet. Today's foreign news, let's start with how the house takes its feet down, literally. Such a miracle was managed in Shanghai, China, where seven thousand six hundred tons of weighty five-story schools were assembled under one hundred and ninety-eight robot feet and walked exactly sixty-one point seven meters with it.	33.0
OWSM 3.1 EBF, peenhäälestatud sünt.)	In today's foreign news, among other things, we show you the oldest professional footballer in the world and the house that was put under the feet. Today's foreign news, we'll start with how the house will take the feet down. In a literally word, such a miracle was managed in China, Shanghai, where seven thousand six hundred tons of weight were considered to be subject to five floors of school, one hundred and ninety-eight robot feet were under the feet and sixty-one point seven meters were walking with him.	25.8

Lisa 4 – ASR hallutsinatsioonid

Allpool on näide automaatse kõnetuvastussüsteemi poolt tagastatud transkriptsioonides sisalduvatest hallutsinatsioonidest ja kordustest. Andmed, millele automaatset kõnetuvastust rakendati, on pärit inglise-eesti tõlkesuuna valideerimisandmestikust.

```
Consecutive repeating lines:

39 consecutive occurrences of 'Boeing is going to be the
   first company in the U.S. to be able to do so.' starting
    from line 187 in file: transkriptsioonid/zdmwtRNJmg8.
   txt

3 consecutive occurrences of 'Yeah.' starting from line 684
    in file: transkriptsioonid/MecVr3Bz4o0.txt

9 consecutive occurrences of 'Yeah.' starting from line
   1015 in file: transkriptsioonid/MecVr3Bz4o0.txt

8 consecutive occurrences of 'I don't know.' starting from
   line 1270 in file: transkriptsioonid/MecVr3Bz4o0.txt

14 consecutive occurrences of 'Yeah.' starting from line
   1296 in file: transkriptsioonid/MecVr3Bz4o0.txt

6 consecutive occurrences of 'Hi, Barbie.' starting from
   line 1310 in file: transkriptsioonid/MecVr3Bz4o0.txt

520 consecutive occurrences of '...' starting from line 344
    in file: transkriptsioonid/DB3bprN1yM8.txt

263 consecutive occurrences of '...' starting from line 866
    in file: transkriptsioonid/DB3bprN1yM8.txt

3 consecutive occurrences of 'TARO KIMURA, BIOJ, After the
   post-war devastation, Japan, at the time, had a pretty
   good economy.' starting from line 1301 in file:
```

transkriptsioonid/DB3bprN1yM8.txt

4 consecutive occurrences of 'The Bank of Japan has ended negative interest rates 17 years since the last.' starting from line 1401 in file: transkriptsioonid/DB3bprN1yM8.txt

3 consecutive occurrences of 'Thank you.' starting from line 1460 in file: transkriptsioonid/DB3bprN1yM8.txt

5 consecutive occurrences of 'We're doing a show.' starting from line 32 in file: transkriptsioonid/3ji8WjdF-nA.txt

223 consecutive occurrences of 'And so it's about a guy who has been cooking salmon in Searle his whole life, cooking his father's dishes, and he just wants to do something different.' starting from line 77 in file: transkriptsioonid/3ji8WjdF-nA.txt

583 consecutive occurrences of 'So this is not a specific German film.' starting from line 306 in file: transkriptsioonid/3ji8WjdF-nA.txt

200 consecutive occurrences of 'NATO Secretary General Jens Stoltenberg.' starting from line 251 in file: transkriptsioonid/0NdjvN3COzg.txt

Lisa 5 – SeamlessM4T eesti-inglise tõlkesuuna BLEU tulemused valideerimisandmestikul

Järgnevalt on kujutatud saadud BLEU skoore eesti-inglise tõlkesuunal, iga valideerimisandmestiku faili kohta eraldi. Mudel, mille jõudlust hinnati on veebi- ja sünteetiliste andmete peal peenhäälestatud SeamlessM4T.

```
Evaluating the file outputs/et/mt/seamlessM4T_v2_large.ft.
    et2ru-en.b/16.12.2020\_-\
    _Tallinna_Linnavalitsuse_kolmapaevane_pressikonverents-
    dGJ9HSmZR8A.et.en.mt in terms of translation quality
    against
data/et/16.12.2020_-
    _Tallinna_Linnavalitsuse_kolmapaevane_pressikonverents-
    dGJ9HSmZR8A.en.OSt
avg sacreBLEU mwerSegmenter 44.926

Evaluating the file outputs/et/mt/seamlessM4T_v2_large.ft.
    et2ru-en.b/aktuaalne-kaamera-ilm-1001-317793.et.en.mt in
     terms of translation quality against data/et/aktuaalne-
    kaamera-ilm-1001-317793.en.OSt
avg sacreBLEU mwerSegmenter 38.141

Evaluating the file outputs/et/mt/seamlessM4T_v2_large.ft.
    et2ru-en.b/aktuaalne-kaamera-ilm-1222-327710.et.en.mt in
     terms of translation quality against data/et/aktuaalne-
    kaamera-ilm-1222-327710.en.OSt
avg sacreBLEU mwerSegmenter 49.991

Evaluating the file outputs/et/mt/seamlessM4T_v2_large.ft.
    et2ru-en.b/aktuaalne-kaamera-ilm-nadal-322248.et.en.mt
    in terms of translation quality against data/et/
    aktuaalne-kaamera-ilm-nadal-322248.en.OSt
avg sacreBLEU mwerSegmenter 41.726

Evaluating the file outputs/et/mt/seamlessM4T_v2_large.ft.
    et2ru-en.b/ringvaade-2033-320571.et.en.mt in terms of
```

```
    translation quality against data/et/ringvaade
    -2033-320571.en.OSt
avg sacreBLEU mwerSegmenter 23.186

Evaluating the file outputs/et/mt/seamlessM4T_v2_large.ft.
    et2ru-en.b/ringvaade-2071-326938.et.en.mt in terms of
    translation quality against data/et/ringvaade
    -2071-326938.en.OSt
avg sacreBLEU mwerSegmenter 23.873

Evaluating the file outputs/et/mt/seamlessM4T_v2_large.ft.
    et2ru-en.b/Valitsuse_pressikonverents__15._oktoober_2020
    -dJypQ9rLypU.et.en.mt in terms of translation quality
    against data/et/Valitsuse_pressikonverents__15.
    _oktoober_2020-dJypQ9rLypU.en.OSt
avg sacreBLEU mwerSegmenter 25.765

Average BLEU: 34.7143
```