Tartu Ülikool

Arvutiteaduse instituut

Informaatika õppekava

Hendrik Püss

Loengute tõlkimine ja subtiitrite genereerimine

Bakalaureusetöö (9 EAP)

Juhendaja(d):

Ardi Tampuu

Tartu 2025

Loengute tõlkimine ja subtiitrite genereerimine

Lühikokkuvõte

Tartu Ülikooli Arvutiteaduse instituudis on märkimisväärne osa õppematerjalidest ja avalikest loengutest kättesaadavad vaid inglise keeles. See seab piirangud nende materjalide ligipääsetavusele eestikeelsele sihtrühmale, kes ei pruugi inglise keelt piisaval tasemel vallata. Käesoleva bakalaureusetöö peamiseks eesmärgiks oli välja töötada tarkvaraline lahendus, mis automatiseerib ingliskeelsete akadeemiliste loengute transkribeerimise, eesti keelde tõlkimise ning ajastatud subtiitrite genereerimise protsessi, vähendades seeläbi vajadust aeganõudva manuaalse töö järele.

Selle eesmärgi saavutamiseks arendati välja rakendus, mis integreerib kaasaegseid tehisintellekti tehnoloogiaid. Rakendus kasutab OpenAI Whisper mudelit automaatseks kõnetuvastuseks (transkribeerimiseks) ning OpenAI ChatGPT API-t genereeritud transkriptsioonide masintõlkeks inglise keelest eesti keelde. Lisaks rakendati moviepy teeki video- ja helifailide töötlemiseks ning Streamlit raamistikku interaktiivse kasutajaliidese loomiseks.

Rakenduse põhifunktsionaalsus hõlmab kolme järjestikust etappi: esiteks loengu heliraja transkribeerimine tekstiks, teiseks saadud ingliskeelse transkriptsiooni tõlkimine eesti keelde ning kolmandaks tõlgitud teksti põhjal täpselt ajastatud subtiitrite genereerimine levinud SubRip (.srt) vormingus. Välja töötatud lahendus pakub praktilist tööriista ingliskeelse akadeemilise sisu kättesaadavuse parandamiseks Eestis.

Võtmesõnad:

Whisper, ChatGPT, Streamlit, moviepy, automaatne transkribeerimine, masintõlge, subtiitrite genereerimine, ligipääsetavus

CERCS:

P176 Tehisintellekt

Lecture Translation and Subtitle Generation

Abstract

At the Institute of Computer Science, University of Tartu, a sizable portion of study materials and public lectures are available only in English. This limits their accessibility for the Estonian-speaking audience who may not possess sufficient English language proficiency. The main objective of this bachelor's thesis was to develop a software solution that automates the process of transcribing English-language academic lectures, translating them into Estonian, and generating timed subtitles, thereby reducing the need for time-consuming manual labor.

To achieve this goal, an application was developed integrating modern artificial intelligence technologies. The application utilizes the OpenAI Whisper model for automatic speech recognition (transcription) and the OpenAI ChatGPT API for machine translation of the generated transcriptions from English to Estonian. Additionally, the moviepy library was employed for processing video and audio files, and the Streamlit framework was used to create an interactive user interface.

The core functionality of the application encompasses three sequential stages: first, transcribing the lecture's audio track into text; second, translating the resulting English transcription into Estonian; and third, generating accurately timed subtitles in the common SubRip (.srt) format based on the translated text. The developed solution offers a practical tool for improving the accessibility of English-language academic content in Estonia.

Keywords:

Whisper, ChatGPT, Streamlit, moviepy, automatic transcription, machine translation, subtitle generation, accessibility

CERCS:

P176 Artificial intelligence

Sisukord

1. Sissejuhatus

Tartu ülikooli arvutiteaduse instituudis, on üha kasvav osa õppematerjalidest ja avalikest loengutest kättesaadavad vaid inglise keeles. See on globaalse teaduskoostöö ja teadmiste leviku seisukohast mõistetav trend, kuid tekitab väljakutseid kohalikul tasandil. Märkimisväärne osa instituudi poolt pakutavatest loengutest ja seminaridest on ingliskeelsed. Selline olukord võib piirata nende materjalide ligipääsetavust eestikeelsele sihtrühmale, eriti üliõpilastele või laiemale arvutiteadusest huvituvale publikule, kes ei pruugi inglise keelt akadeemilise sisu mõistmiseks piisavalt vallata.

Keelebarjäär võib mõjutada õppeprotsessi tõhusust ja teadmiste omandamise sügavust. Uuringud on näidanud, et võõrkeeles õppimine võib suurendada kognitiivset koormust ning teatud juhtudel võivad õpitulemused olla madalamad võrreldes emakeelse õppega. Näiteks, üks Türgi ülikoolis läbi viidud uuring leidis, et õpetamine emakeelest erinevas keeles (antud juhul inglise keeles Türgi üliõpilastele) mõjutas negatiivselt üliõpilaste akadeemilist edukust, mida mõõdeti semestri keskmise hindega [1]. Seega on oluline otsida lahendusi, mis aitaksid kaasa kvaliteetse akadeemilise sisu kättesaadavuse parandamisele emakeeles, toetades seeläbi võrdseid võimalusi hariduse omandamisel ja teadmiste levikul Eestis.

Käesoleva bakalaureusetöö peamiseks eesmärgiks oli välja töötada tarkvaraline lahendus, mis automatiseerib ingliskeelsete akadeemiliste loengute transkribeerimise, eesti keelde tõlkimise ning ajastatud subtiitrite genereerimise protsessi. Sellise automatiseeritud süsteemi loomine aitab

vähendada märkimisväärselt aeganõudvat ja kulukat manuaalset tööd, mis on traditsiooniliselt seotud loengumaterjalide tõlkimise ja subtiitrimisega.

Selle eesmärgi saavutamiseks integreeriti rakendusse kaasaegseid tehisintellekti tehnoloogiaid. Automaatseks kõnetuvastuseks (transkribeerimiseks) kasutati OpenAI Whisper mudelit, mis on tuntud oma võimekuse poolest töödelda suurt hulka mitmekeelseid andmeid ning tagada kvaliteetne transkriptsioon ka keerulistes akustilistes tingimustes [2, 3]. Genereeritud ingliskeelsete transkriptsioonide masintõlkeks eesti keelde rakendati OpenAI ChatGPT API-t, täpsemalt GPT-4o mini mudelit , mis pakub head tasakaalu tõlkekvaliteedi ja kuluefektiivsuse vahel [4, 5]. Video- ja helifailide töötlemiseks, sealhulgas heli eraldamiseks ja subtiitrite videole põimimiseks, kasutati Pythoni teeki MoviePy. Interaktiivse ja kasutajasõbraliku veebipõhise kasutajaliidese loomiseks valiti Streamlit raamistik [6].

Rakenduse põhifunktsionaalsus realiseeriti kolmeastmelise töövoona: esmalt transkribeeritakse loengu helirada tekstiks, seejärel tõlgitakse saadud ingliskeelne transkriptsioon masintõlke abil eesti keelde ning viimase sammuna genereeritakse tõlgitud teksti põhjal subtiitrid SubRip (.srt) vormingus. Välja töötatud lahendus pakub praktilist tööriista ingliskeelse akadeemilise sisu kättesaadavuse parandamiseks Eestis, aidates kaasa teadmiste laiemale levikule ja eestikeelse terminoloogia arengule.

ChatGPT kasutamise avaldus: Käesoleva bakalaureusetöö koostamisel kasutati OpenAI keelemudelit ChatGPT mitmes etapis. Seda rakendati ideede genereerimisel ja struktureerimisel, teoreetilise tausta ning sissejuhatuse osade kavandamisel, abivahendina Pythoni koodilõikude kirjutamisel ja silumisel (enamasti seoses MoviePy ja Streamlit teekidega), eestikeelse teksti õigekirja ja stiili kontrollimisel ning akadeemilise sõnastuse parandamisel. Kogu ChatGPT poolt genereeritud või muudetud sisu on autori poolt kriitiliselt üle vaadatud, kohandatud ja kinnitatud, et tagada töö originaalsus ja akadeemiline korrektsus.

2. Teoreetiline taust

Antud peatükk kirjeldab loengute automaatse transkribeerimise, nende eesti keelde tõlkimise ja seejärel subtiitrite loomise teoreetilisi aluseid. See süveneb peamistesse kasutatavatesse tehnoloogiatesse: automaatsesse kõnetuvastusse (Automatic Speech Recognition, ASR), masintõlkesse (Machine Translation) ja subtiitrite loomise tehnikatesse, tuues välja selle uurimistööga seotud olulised mõisted, mudelid, väljakutsed ja hindamismetoodikad.

2.1 Automaatne kõnetuvastus

Automaatne kõnetuvastus (Automatic Speech Rrecognition) on valdkond arvutiteaduses ja keeletehnoloogias, mis keskendub meetodite ja tehnoloogiate arendamisele, mis võimaldavad arvutitel ära tunda ja tõlkida suulist keelt tekstiks. ASR-süsteemi peamine eesmärk on akustiline kõnesignaal täpselt vastavaks tekstiliseks esituseks muuta [7].

2.1.1 Automaatse kõnetuvastuse areng

Ajalooliselt toetusid automaatse kõnetuvastuse süsteemid suuresti peidetud Markovi mudelitele (Hidden Markov Model) kombineerituna Gaussi segu mudelitega (Gaussian Mixture Model) akustiliseks modelleerimiseks ning n-gramm mudelitele keele modelleerimiseks [8]. Kuigi need lähenemised olid teatud ülesannete puhul edukad, nõudsid need tihti eraldi komponente ja märkimisväärset keeleteaduslikku ekspertiisi. Süvaõppe areng on valdkonna revolutsiooniliselt muutnud, viies närvivõrkudel põhinevate lähenemiste domineerimiseni [8].

Otsast-lõpuni (ingl k end-to-end, E2E) mudelid, nagu need, mis põhinevad konnektsionistlikul ajalisel klassifitseerimisel (Connectionist Temporal Classification), rekurrentsetel närvivõrkude transduktoritel (Recurrent Neural Network Transducer) ja tähelepanul põhinevatel järjestusest-järjestusse arhitektuuridel (nt Listen, Attend and Spell; LAS), on muutunud üha levinumaks. Need mudelid püüavad otse kaardistada akustilisi tunnuseid tähtede või sõnade järjestusteks, lihtsustades töövoogu ja saavutades tihti parema jõudluse [9, 10].

Transformeril põhinevad arhitektuurid, mis algselt töötati välja loomuliku keele töötluse (natural language processing) ülesannete jaoks, on samuti edukalt kohandatud ASR jaoks, näidates tugevat jõudlust, eriti suurtel andmestikel [3, 14].

2.1.2 Väljakutsed loengute transkribeerimisel

Loengute transkribeerimine esitab järgmiseid väljakutseid:

- **Kõneleja varieeruvus:** Üliõpilaste küsimused toovad kaasa varieeruvuse.
- **Akustilised tingimused:** Ruumi akustika (kaja), taustamüra (publiku sosin, kütte- ja ventilatsioonisüsteemid, välised helid) ning mikrofoni kvaliteet/paigutus mõjutavad oluliselt heli kvaliteeti.
- **Spontaanne kõne:** Loengud sisaldavad nähtusi nagu kõnehäired (kõhklused, täitesõnad nagu "hmm", kordused) ja grammatilised ebajärjekindlused, mis on ASR-süsteemidele raskemini käsitletavad kui planeeritud kõne.
- **Valdkonnapõhine sõnavara:** Loengud sisaldavad sageli tehnilisi termineid, erialasõnavara, akronüüme ja pärisnimesid, mis on spetsiifilised antud teemale ning mida ei pruugi esineda automaatse kõnetuvastuse süsteemi üldises keelemudelis, nõudes valdkondlikku kohandamist.

2.1.3 Hindamismõõdikud

Standardne meetod automaatse kõnetuvastuse (Automatic Speech Recognition) tulemuslikkuse hindamiseks on sõna veamäär (WER - Word error rate), mis mõõdab vigade protsenti (asendused, lisandused, kustutused) võrreldes õige transkriptsiooniga. Tähemärgi veamäära (CER - Character error rate) kasutatakse mõnikord, eriti morfoloogiliselt rikaste keelte puhul või kui keskendutakse tähemärkide tasandi täpsusele [3].

2.2 Masintõlge

Masintõlge (Machine Translation) on arvutuslingvistika valdkond, mis uurib tarkvara kasutamist teksti või kõne tõlkimiseks ühest keelest teise. Eesmärk on luua tõlge, mis on nii tähenduselt täpne kui ka sihtkeeles ladus.

2.2.1 Masintõlke Areng

Varased masintõlkesüsteemid olid peamiselt reeglipõhised (Rule-based machine translation), tuginedes käsitsi loodud keelereeglitele ja sõnastikele. Statistiline masintõlge (Statistical machine translation), eriti fraasipõhine stastiline masintõlge, muutus hiljem domineerivaks, õppides tõlkimistõenäosusi suurtest paralleelkorpustest [11].

Praegune tipptehnoloogia on neuraalne masintõlge (Neural machine translation), mis kasutab süvaõppe võrke, tavaliselt järjestuselt-järjestusele mudeleid koos tähelepanumehhanismidega [12, 13].

Transformeri mudelitest on saanud de facto standardarhitektuur neuraalse masintõlke jaoks tänu nende võimele käsitleda pikki sõltuvusi ja paralleliseerimise võimalusi, mis on viinud märkimisväärsete edasiminekuteni tõlkekvaliteedis [14].

2.2.2 Eesti keelde tõlkimise väljakutsed

Transkribeeritud loengumaterjali tõlkimine eesti keelde tekitab spetsiifilisi raskusi:

- **Morfoloogiline keerukus:** Eesti keel on soome-ugri keel, millel on rikkalik morfoloogia (aglutinatsioon ja käänamine), hõlmates arvukaid käändeid ja keerukaid sõnamoodustusi. See muudab grammatiliselt täpse ülekandmise morfoloogiliselt lihtsamatest lähtekeeltest (nagu inglise keel) masintõlkesüsteemidele keeruliseks [15].
- **Sõnade järjekord:** Kuigi eesti keeles on võrreldes inglise keelega suhteliselt paindlik sõnade järjekord, võib tähenduse nüansse edasi anda sõnade järjekorra valikutega, mida MT-süsteemid ei pruugi suuta loomulikult jäljendada.
- **Ressursside vähesus:** Kuigi edusamme on tehtud, on eesti keele jaoks üldiselt vähem paralleelseid korpuseid NMT mudelite treenimiseks võrreldes suuremate maailma

keeltega, mis võib mõjutada tõlkekvaliteeti, eriti akadeemiliste loengute spetsiifilistes valdkondades.

- **Vigade levik:** ASR-i transkriptsioonifaasist pärinevad vead (valed sõnad, valesti tuvastatud terminid) võivad MT väljundis võimenduda või viia mõttetute tõlgeteni.

2.3 Subtiitrite genereerimine

Subtiitrite genereerimine tähendab selle projekti kontekstis ajastatud tekstisegmentide loomist, mis vastavad tõlgitud eestikeelse loengu sisule. Peamine väljakutse seisneb pideva tõlgitud teksti segmenteerimises ning sobivate algus- ja lõpuaja templite määramises, et tagada subtiitrite ilmumine sünkroonis algse loengu video kõnega.

2.3.1 Eesmärk ja vormingud

Subtiitritel on mitu olulist funktsiooni. Esiteks parandavad need juurdepääsetavust, muutes audiovisuaalse sisu kättesaadavaks kurtidele või vaegkuuljatele, nagu nõuavad juurdepääsetavuse standardid nagu Veebisisu juurdepääsetavuse suunised (Web Content Accessibility Guidelines) [16]. Samuti aitavad need mõista sisu inimestele, kes antud keelt emakeelena ei räägi, mürarikkas keskkonnas või keerulise terminoloogia puhul. Lisaks hõlbustab ajastatud tekst sisu otsitavust ja navigeerimist, sest osad keskkonnad võimaldavad subtiitrite alusel otsida videoid või video sees.

Subtiitrite salvestamiseks on olemas mitu tekstipõhist vormingut. Kõige levinumate hulgas on SubRip (.srt), mis on tuntud oma lihtsuse poolest [17], ja WebVTT (.vtt), mis on W3C poolt standardiseeritud kaasaegsem vorming, pakkudes rikkalikumaid funktsioone veebipõhise videosisu jaoks [18]. Mõlemad vormingud salvestavad põhimõtteliselt tekstisegmentide jadasid koos nende täpse kuvamise ajastusinfoga (algus- ja lõpuajad).

2.3.2 Automaatne subtiitrite genereerimine

Subtiitrite automaatne genereerimine tõlgitud tekstist, eriti kui algne ajastus pärineb teisest keelest, tekitab märkimisväärseid väljakutseid. Protsess hõlmab tavaliselt kahte peamist sammu: segmenteerimist ja ajastamist.

2.3.3 Segmenteerimine

Tõlgitud eestikeelne tekst, mis on algselt pidev tekstiplokk, tuleb jagada loogilisteks osadeks, mida sageli nimetatakse subtiitrifraasideks. Tõhusa segmenteerimise eesmärk on viia need fraasid vastavusse loomulike pauside või lausepiiridega algses kõnes, järgides samal ajal rangelt kehtestatud loetavuspiiranguid. Need piirangud, mida sageli kirjeldatakse valdkonna stiilijuhistes, hõlmavad tavaliselt tähemärkide arvu rea kohta ja ridade arvu subtiitrifraasi kohta (nt maksimaalselt kaks rida), et tagada vaatajatele teksti mugav lugemine selle kuvamise kestuse jooksul [19]. Loomuliku keele töötluse ja kõnetöötluse uuringud uurivad erinevaid tehnikaid selle segmenteerimise tõhusaks automatiseerimiseks [20].

2.3.4 Ajastamine

Igale tõlgitud subtiitrifraasile täpse ajastuse määramine on sünkroniseerimiseks oluline. Iga fraas nõuab täpset algus- ja lõpuaega, mis vastab algsele kõneldud sisule loengu audio/video ajajoonel. Tõlke sõnade arv ja struktuur erinevad lähtekeelest. See nõuab tõlgitud teksti joondamist tagasi lähteheli ajajoonele. Sellised tehnikad nagu sundjoondamine, mis kasutavad sageli automaatse kõnetuvastuse (automatic speech recognition) mudeleid, on levinud uurimis- ja kasutusvaldkond, et kaardistada tekstisegmente (kas algseid või tõlgitud) konkreetsete heliaja intervallidega [21]. Ajastusmehhanismid vormingutes nagu WebVTT on loodud toetama sellist täpsustaset [18].

2. Kasutatud tehnoloogiad

Peatükis on ülevaade rakenduse loomisel kasutatud tehnoloogiatest. Erinevad tehnoloogiad on eraldi alapeatükkides. Igas alapeatükis antakse ülevaade sellest, mis kujul ja kuidas vastavat tehnoloogiat rakenduse loomisel kasutati.

2.1 Töövoog

Teoreetilises taustas kirjeldatud moodulid hõlmavad töövoogu vajaliku rakenduse valmimiseks: ASR (automaatne kõnetuvastus) teisendab kõne tekstiks, MT (masintõlge) tõlgib selle teksti ning subtiitrite genereerimine lisab ajastuse informatsiooni. Vigade tekkimine varases etapis levib paratamatult järgmistesse etappidesse. Halvasti transkribeeritud sõna võib viia vale tõlkeni, mis seejärel ajastatakse ja kuvatakse subtiitrina. Teadusuuringud käsitlevad nii modulaarseid lähenemisviise (iga komponendi eraldi optimeerimist) kui ka otse-lõpuni süsteeme, mis võivad teisendada kõne otse ajastatud ja tõlgitud tekstiks. See võib vähendada vigade levimist, kuid suurendab mudeli keerukust. Valik nende lähenemisviiside vahel sõltub andmete kättesaadavusest, arvutusressurssidest ja soovitud jõudlusomadustest [22].

2.2 Moviepy

MoviePy on Pythoni teek videotöötluseks, mida saab kasutada põhilisteks operatsioonideks nagu lõikamine, konkateneerimine, tiitrite lisamine, video komponeerimine, helitöötlus jne [22]. Antud rakenduses kasutatakse MoviePy-d mitmes etapis:

Heli eraldamine videost: Funktsioon *extract_full_audio* kasutab *VideoFileClip(video_path).audio.write_audiofile(audio_path*) meetodit, et eraldada video helirada.

- **Sisend:** Üleslaetud videofail, nt temp_processing/kasutaja_video.mp4.
- **Väljund:** Eraldatud helifail, nt *temp_processing/kasutaja_video_full_audio.wav*.

Heli tükeldamine lõikudeks: Funktsioon split_audio_into_chunks kasutab *AudioFileClip(audio_path).subclip(start_time, end_time).write_audiofile(chunk_path)*, et jagada pikk helifail lühemateks, hallatavateks lõikudeks.

- **Sisend:** Eraldatud täispikk helifail, nt *temp_processing/kasutaja_video_full_audio.wav*.
- **Väljund:** Nimekiri helilõikude failinimedest ja nende algusaegadest, nt *[('temp_processing/audio_chunks/audio_chunk_1.wav', 0, 120), ('temp_processing/audio_chunks/audio_chunk_2.wav', 120, 240), ...]*. Need .wav failid on sisendiks OpenAI Whisperile.

Subtiitrite põimimine videoga: Funktsioon *embed_subtitles()* kasutab *CompositeVideoClip* ja *SubtitlesClip*, et lisada tõlgitud subtiitrid videole.

- **Sisend:** Originaal videofail (nt *temp_processing/kasutaja_video.mp4*) ja kombineeritud tõlgitud .srt fail (nt *output_processed/kasutaja_video_Estonian_subtitles.srt*).
- **Väljund:** Uus videofail koos põimitud subtiitritega, nt *output_processed/kasutaja_video_with_Estonian_subs.mp4*.

2.3 OpenAI Whisper

OpenAI Whisper on automaatne kõnetuvastussüsteem (inimkõne sisu automaatne äratundmine), mis on treenitud 680 000 tunni jooksul veebist kogutud mitmekeelsete andmete põhjal ja Whisperiga saab transkribeerida (helist tekstikujule viia) erinevaid keeli [2]. Samuti võimaldab Whisper oma mudeleid indiviididel kasutada, kuna need on avatud lähtekoodiga (MIT Litsents). See toetab arendajate võimalusi ehitada kasulike rakendusi ja viib edasi uuringuid keeletehnoloogias [2].

Whisper transkribeerib tekste, teisendades helisignaali tekstiks ning lisab automaatselt kirjavahemärgid ja lõigujaotused, et lihtsustada transkriptsiooni loetavust. Lisaks sellele sisaldab Whisperi tehnoloogia mitmeid optimeerimisi, näiteks see suudab eristada erinevaid kõnelejaid ja töötab hästi ka keerulistes akustilisistes tingimustes, nagu taustamüra või madala kvaliteediga salvestised. Whisper toetab ka mitmeid erinevaid keeli, aga antud lõputöö kontekstis huvitab meid vaid inglise keele transkribeerimine [22].

Whisper normaliseerib transkribeeritud inglise keelsed tekstid standardiseeritud kujule. Näiteks, tuvastab numbrite väljendusi ja valuutasid, ja kasutab sõnade asemel araabia numbreid, .s.t “Ten thousand dollars“ -> “$10000”. Samuti eemaldab järgmised sõnad: hmm, mm, mhm, mmm, uh, um. Peale selle on ka palju teisi standardiseerimisreegleid [3].

Avatud lähtekoodiga Whisperil on kuus mudeli suurust, millest neli on ainult inglise keelsed variandid. Mudeleid eristab kiirus ja täpsus [3]. Antud töös kasutati inglise keelset mudelit *medium.en*, kuna tegemist on kõige täpsema inglise keelele spetsialiseeritud transkribeeriva mudeliga mida Whisper pakub.

Mudel laeti alla kohalikku Python virtuaalsesse keskkonda Whisper GitHubi repositooriumist. Mudelil kasutati funktsionaalsust *transcribe*, millesse laeti helifail (rakenduses on see helilõik, näiteks *temp_processing/audio_chunks/audio_chunk_1.wav*). Mudel transkribeerib helifailist transkriptsiooni. Funktsiooni *transcribe_with_whisper* väljund on Pythoni sõnastik, mis sisaldab transkribeeritud teksti segmente koos ajatemplitega. Näiteks:

```
{
  'text': ' Hello world, this is a test.',
  'segments': [
    {'id': 0, 'seek': 0, 'start': 0.50, 'end': 1.80, 'text': ' Hello world,'},
    {'id': 1, 'seek': 0, 'start': 2.00, 'end': 3.50, 'text': ' this is a test.'}
  ],
  'language': 'en'
}
```

Joonis 2.3.1: Whisperi väljund

Seda sõnastikku kasutatakse funktsioonis *process_audio_chunk*, et luua helilõigule vastav originaalkeelne .srt fail (subtiitrifail). Selle vahesammu väljundiks on näiteks fail *output_processed/chunk_srts/audio_chunk_1_original.srt*, mille sisu on:

```
1
00:00:00,500 --> 00:00:01,800
Hello world,

2
00:00:02,000 --> 00:00:03,500
this is a test.
```

Joonis 2.3.2: SRT väljund

Need individuaalsed .srt failid töödeldakse edasi järgmistes alapeatükkides käsitletud tehnoloogiatega (täpsemalt ChatGPT API-ga tõlkimiseks ja seejärel MoviePy-ga kombineerimiseks).

2.4 ChatGPT API ja ChatGPT 4o mini

ChatGPT API on avalik ja tasuline rakendusliides, mis võimaldab kasutajatel enda programmi sees kasutada OpenAI tehisintellekti mudeleid. See pakub enamasti huvi arendajatele, kes saavad OpenAI mudelitega uusi rakendusi luua. [4]

ChatGPT API kaudu saab muuhulgas kasutada ChatGPT 4o mini mudelit enda valitud sisendiga programmi sees. Võimalik on järelikult kasutada seda mudelit transkribeeritud teksti eesti keelde tõlkimiseks, andes sisendiks teksti, mida tõlkida soovitakse. Nii saab tehisintellekt teha kontekstuaalselt parima tehistõlke.

Madalate kulude ja latentsusajaga sobib GPT-4o mini hulga erinevate ülesannete töötlemiseks, näiteks rakendused, mis kutsuvad mitut rakendusliidest ja edastavad mudelile suure hulga konteksti või vestluste ajalugu [5]. Suure hulga konteksti töötlemine on antud lõputöös kõige asjakohasem, kuna tegemist on suure hulga teksti töötlemisega, mis on transkribeeritud loengutest. Samuti ka rakenduse loomiseks ette nähtud madala eelarve korral on madalaima kuluga mudel kõige sobilikum.

Mudelit kasutati subtiitrite tõlkimises. Rakenduse koodis funktsioon *translate_subtitles* võtab sisendiks eelmises etapis loodud .srt faili (näiteks *output_processed/chunk_srts/audio_chunk_1_original.srt*) sisu, mis on parsitud Pythoni listiks

sõnastikest. Mudelile anti süsteemne käsklus (*prompt*): "You are a professional translator specialized in subtitle translation to {*target_language*}. Translate the following subtitles, preserving the context and nuance across all entries. Provide a list of translations that match the input order exactly.", kus objekt *target_language* on funktsiooni parameeter (antud töö kontekstis „Estonian“). Koodis antakse sisendiks umbes 300 sõna. Kasutaja sisendiks API-le on nummerdatud nimekiri transkribeeritud tekstidest. Näiteks, kui sisendiks on:

```
1. Hello world,
2. this is a test.
```

Joonis 2.4.1: Transkribeeritud teksti sisend

siis API väljastab tõlgitud read. Funktsioon *translate_subtitles()* töötleb selle vastuse ja asendab originaaltekstid tõlgitud tekstidega, säilitades ajakoodid. Selle vahesammu väljundiks on tõlgitud subtiitritega .srt fail vastava helilõigu kohta, näiteks *output_processed/chunk_srts/audio_chunk_1_translated.srt*, mille sisu võib olla:

```
1
00:00:00,500 --> 00:00:01,800
Tere maailm,

2
00:00:02,000 --> 00:00:03,500
see on test.
```

Joonis 2.4.2: Tõlgitud SRT väljund

Seejärel need individuaalsed tõlgitud .srt (koodis 120 sekundit) failid kombineeritakse üheks terviklikuks subtiitrifailiks.

2.5 Streamlit

Streamlit on Pythonil põhinev avatud lähtekoodiga raamistik, mis võimaldab arendajatel kiiresti luua ja jagada interaktiivseid veebirakendusi, eriti andmeteaduse ja masinõppe projektide jaoks [6]. See võimaldab Pythoni skriptidest luua kasutajaliideseid ilma keerulise esi- ja tagaliidese eraldi arendamiseta.

Antud rakenduse puhul oli Streamlit valitud kogu interaktiivse veebirakenduse loomiseks. Rakendus võimaldab kasutajal otse veebiliidese kaudu üles laadida videofaili. Streamlit käivitab seejärel taustal Pythoni koodi, mis teostab vajaliku andmetöötluse ja funktsionaalsuse. Pärast töötlemist esitab Streamlit kasutajale otse liideses allalaadimisvõimaluse subtiitrite faili jaoks (formaadis .srt) ja/või videofaili jaoks (formaadis .mp4).

Streamlit oli sobilik valik rakenduse loomiseks, kuna see lihtsustab oluliselt interaktiivse kasutajaliidese loomist Pythoni-põhisele protsessile ning ühildub suurepäraselt teiste projektis kasutatud Pythoni teekidega, mis tagavad rakenduse põhifunktsionaalsuse. Töös kasutati Streamlit versiooni 1.37.1, mis oli rakenduse arendamisel uusim väljalase.

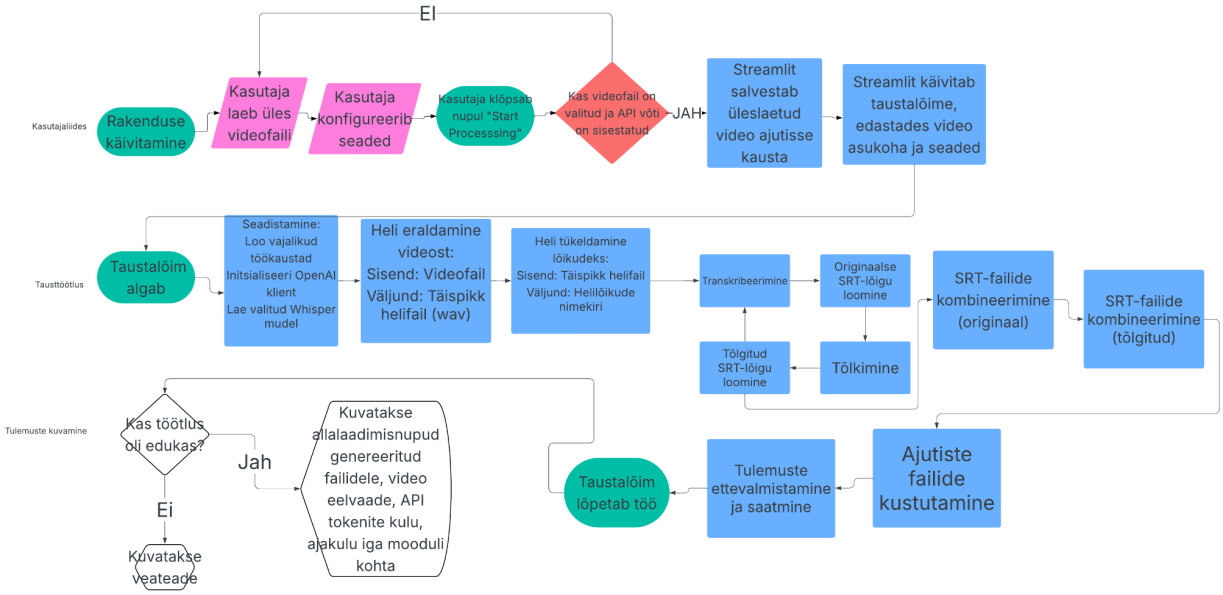
2.6 Süsteemiarhitektuur

Rakendus on loodud modulaarse töövoona (pipeline), mida juhib keskne Pythoni skript (main_app.py). See kasutab Streamliti teeki, et pakkuda veebipõhist graafilist kasutajaliidest (Graphical User Interface), mis võimaldab kasutajatel hõlpsasti videofaile üles laadida, töötlemisparameetreid konfigureerida ja edenemist jälgida.

Tagamaks, et kasutajaliides jääks reageerivaks potentsiaalselt pikkade töötlemisaegade jooksul, käivitatakse peamine videotöötlusloogika (process_video_threaded) eraldi lõimes (thread). Suhtlus töötlemislõime ja Streamliti kasutajaliidese vahel edenemise uuenduste ja tulemuste edastamiseks toimub Pythoni Queue objektide (progress_queue, result_queue) abil.

Arhitektuur tugineb paljudele abifunktsioonidele, millest igaüks vastutab konkreetse ülesande eest töövoos (nt heli eraldamine, transkribeerimine, tõlkimine, subtiitrite genereerimine). Selline modulaarne disain soodustab koodi taaskasutatavust ja hooldatavust. Vaheetappidel genereeritud

ajutised failid (nagu heliklipid) salvestatakse spetsiaalsesse temp_processing kausta, samas kui lõplikud väljundid salvestatakse output_processed kausta.



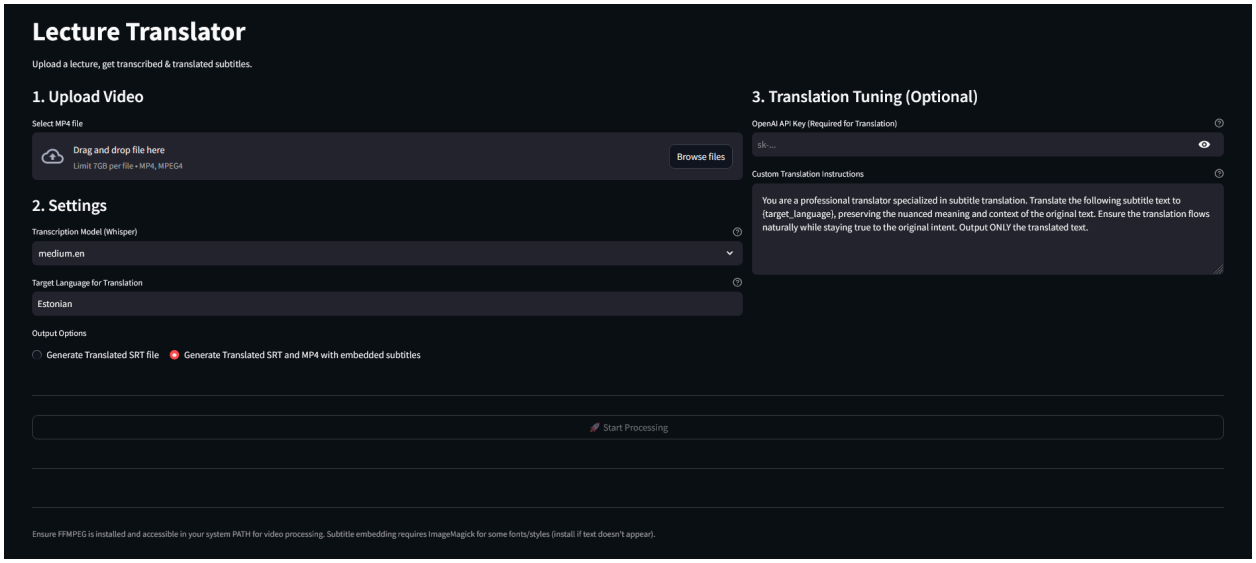
Joonis 2.6.1: Süsteemi arhitektuuri flowchart

3. Tulemused

Selles peatükis esitatakse loodud rakenduse testimise tulemused, analüüsitakse erinevate moodulite edukust, jõudlust, resurssikulu ning antakse hinnang kogu süsteemi väljundile.

3.1 Kasutajaliides

Rakenduse kasutajaliides loodi Streamliti abil, pakkudes lihtsat ja intuitiivset viisi videofailide üleslaadimiseks, parameetrite seadistamiseks ning tulemuste allalaadimiseks. Kasutajaliides võimaldab valida transkribeerimise mudeli, sisestada OpenAI API võtme, määrata sihtkeele tõlkimiseks ning kohandada tõlkeprompti.



Joonis 3.1.1: Rakenduse graafiline kasutajaliides

Kasutajaliides kuvab töötlemise ajal progressiriba ja teateid jooksvate operatsioonide kohta, mis on oluline tagasiside andmiseks pikemate protsesside puhul. Pärast edukat töötlemist ilmuvad allalaadimisnupud genereeritud SRT-failidele ja/või subtiitritega videofailile.

3.2 Moodulite edukus ja jõudlus

Järgnevalt analüüsitakse rakenduse peamiste moodulite (heli eraldamine ja tükeldamine, transkribeerimine, tõlkimine, subtiitrite kombineerimine ja põimimine) edukust ning nendele kuluvat tööaega. ChatGPT APIga tõlkimisel tuuakse välja ka rahaline kulu. Testimiseks kasutati andmeteaduse seminari pikkusega 02:50:53. Antud seminar valiti, sest sellel on mitu külalisesinejat erinevate kõnemaneeridega, seminar on teostatud seminariruumis loengutele kohase helikvaliteediga. Programmi tööd teostati süsteemis, mille spetsifikatsioonid on järgmised:

CPU: 11th Gen Intel(R) Core(TM) i5-11300H @ 3.10GHz   3.11 GHz

RAM: 16,0 GB (15,7 GB usable)

GPU: NVIDIA GeForce RTX 3050 Ti Laptop GPU

3.2.1 Heli eraldamine ja tükeldamine (MoviePy)

Selles etapis on peamine edukuse kriteerium see, et helirada eraldatakse videost korrektselt ja täispikkuses ning seejärel tükeldatakse see vigadeta vastavalt CHUNK_DURATION parameetrile (koodis 120 sekundit).

02:50:53 pikkuse video (10253 sekundit) puhul eraldati helifail *video_full_audio.wav*. Seejärel tükeldati see 10253s/120s≈85.44 lõiguks, mis tähendab 86 helilõiku (85 täispikka 120-sekundilist lõiku ja üks lühem lõpulõik).

- Arvutusajad:
    - Heli eraldamine (Audio extraction): 46,85 sekundit.
    - Eraldatud helifaili tükeldamine (Audio splitting): 53,46 sekundit

3.2.2 Transkribeerimine (OpenAI Whisper)

Transkribeerimise kvaliteeti saab hinnata kvantitatiivselt Word Error Rate (WER) mõõdikuga, mis võrdleb masina genereeritud transkriptsiooni ideaalse, inimese poolt loodud transkriptsiooniga. WER arvutatakse järgmise valemiga [23]:

$$WER = \frac{(S + D + I)}{N}$$

,

Joonis 3.2.2.1: WER Valem [23]

kus S on asenduste arv (substitutions), D kustutamiste arv (deletions), I lisamiste arv (insertions) ja N sõnade arv inimese poolt koostatud transkriptsioonis. Madalam WER näitab paremat täpsust [23].

WER hindamiseks kasutati suvalist 120-sekundilist helilõiku loengust. Kasutati *audio_chunk_7_original.srt*. Helilõik algab täispikas loengus alates 12:00. Antud helilõigu transkriptsioon hinnati WER valemi järgi ja tulemused olid järgmised:

Word Error Rate (WER): 4/205≈0,02

Asenduste arv (S): 0

Kustutamiste arv (D): 1

Lisamiste arv (I): 3

Sõnade arv inimese poolt koostatud transkriptsioonis (N): 205

WER on väga madal antud lõigu kohta ja osad, kus tekkisid vead, olid pigem Whisper poolt tehtud parandused kõnemaneerile. Näiteks lõigus “... and there shouldn’t be a case where two, a pair of speakers only occurs …”, tegi Whisper transkriptsiooni “ … and there shouldn’t be a case where a pair of speakers only occurs …”. Eemaldas “two”, mille rääkija parandas “pair” vastu kuna tegemist oli ebavajaliku sõnaga transkriptsioonis.

Autori poolt on antud väga hea hinnang antud videolõigu Whisperi poolt loodud transkriptsioonile. Tegemist on rääkijaga, kes kasutab jutu sees täitesõnu (filler words) näiteks “ee”, “ää”, “hmm” ja Whisper tuvastas need ära ja ei lisanud neid transkriptsiooni. Rääkijal on ka kohati tugev aktsent, mis ei tekitanud Whisperile probleeme.

Whisperil tekib probleeme enamasti Eesti nimedega ja kohanimedega. Kuna tegemist on inglise keelt tuvastava mudeliga (medium.en), siis sel pole või on vähe andmeid Eesti nimede ja kohanimede kohta. Näiteks “Meelis” transkribeeriti “Meili” ning “Tallinn” transkribeeriti “Italian”. Tegemist on kõla poolest sarnaste sõnadega (kui on öeldud inglise keelses häälduses), aga ühe sõna jaoks võib olla rohkem andmeid kui teise.

- Arvutusajad (86 helilõiku, kasutades RTX 3050 Ti Laptop GPU-d):
    - Whisper mudeli laadimine: 10,65 sekundit.
    - Kõigi 86 helilõigu transkribeerimine kokku (Total transcription time chunks): 9207.12 sekundit (ligikaudu 2 tundi, 33 minutit ja 27 sekundit).
        - See teeb keskmiselt 9207.12s/86≈107,06 sekundit ühe 120-sekundilise (või lühema viimase) helilõigu kohta.

3.2.3 Tõlkimine (ChatGPT API - GPT-4o mini)

Tõlkekvaliteeti hinnati kvalitatiivselt, analüüsides tõlke adekvaatsust, grammatilist korrektsust ja konteksti säilimist. Hindamine põhines juhuslikult valitud lõigul (*audio_chunk_25*) ning üldisel muljel kogu tõlgitud materjalist.

Adekvaatsus: Tõlge edastab algteksti mõtte. Kõik laused on sisuliselt korrektselt tõlgitud, säilitades algse sõnumi. Näiteks "The relationship between synthetic data and AI in a way is bi-directional" on tõlgitud kui "Suhe sünteetiliste andmete ja tehisintellekti vahel on oma moodi kaksisuunaline," mis tabab hästi ära nii terminoloogia kui ka nüansi "in a way" ("oma moodi"). Samuti on "roadblocks" tõlgitud asjakohaselt kui "takistused".

Grammatiline korrektsus: Tõlge on enamasti grammatiliselt korrektne. Lausestus on loogiline, kuid tekivad raskused eesti keele käänetega, näiteks “And that can be very hard to achieve” tõlgiti “Ja see võib olla väga raske saavutada”. Korrektsem oleks “Ja seda võib olla väga raske saavutada”. Samas on tõlge lugeja jaoks jätkuvalt mõistetav. Enamus käändeid on korrektsed antud tekstis kuid esineb sarnaseid vigu.

Konteksti säilimine ja loomulikkus: Tõlge arvestab konteksti ning kõlab eesti keeles loomulikult ja sujuvalt. See ei mõju kohmaka otsetõlkena, vaid pigem nii, nagu oleks tekst

algselt eesti keeles koostatud. Näiteks "You had this nice pipeline of what the kind of way with six steps could be towards putting it into use" on tõlgitud ladusalt: "Teil oli see tore töövoog kuue sammuga, mis võiks olla tee selle kasutusse viimiseks." Väljend "töövoog" on "pipeline" jaoks hea vaste selles kontekstis. Ka küsimused nagu "have you looked at all yet with Cybernetica into the AI methods" on tõlgitud loomulikult: "kas olete Cybernetica juures vaadanud tehisintellekti meetodeid".

Autori poolt on antud rahuldav hinnang kogu teksti tõlkele. Käänetega esineb vigu, kuid tõlge on enamasti kohane ja ladus. Kogu loengu peale esines üksikuid lõike, kus mõni rida lõpus jäi tõlkimata näiteks lõigus audio_chunk_21 on kaks viimast rida tõlkimata jäetud. Selle tekitab olukord kui tõlkemoodul võtab kokku inglise keeles lühemad tekstilõigud ühte ritta. Et seda vältida peaks katsetama erinevat prompti, või anda võimalus kasutajale subtiitreid üle vaadata.

- Arvutusajad ja API kulu (konkreetse testi põhjal, 86 helilõiku/API päringut):
    - OpenAI kliendi initsialiseerimine: 0.26 sekundit.
    - Kõigi 86 helilõigu subtiitrite tõlkimine kokku (Total translation time chunks): 753.53 sekundit (ligikaudu 12 minutit ja 34 sekundit).
    - See teeb keskmiselt 753.53s/86≈8.76 sekundit ühe lõigu subtiitrite tõlkimiseks.
    - API Kulu (konkreetse testi põhjal):
        - API päringuid kokku: 86.
        - Sisendi tokenid (prompt tokens): 37484.
        - Väljundi tokenid (completion tokens): 39798.
        - Tokeneid kokku: 77282.
        - Hinnanguline kulu GPT-4o mini mudeliga (hinnad: $1.1/miljon sisendtookenit, $4.4/miljon väljundtookenit seisuga mai 2025)[24]:
            - Sisendi hind: $(37484/1000000) x $1.1 ≈ $0,04
            - Väljundi hind: $(39798/1000000) x $4.4 ≈ $0,18
            - Kogu tõlkekulu testitud seminarile: $0,04 + $0,18 = $0,22 (ligikaudu 20 eurosenti, eeldades 11.05.2026 kurssi 1 EUR = 1,13 USD)

3.2.4 Subtiitrite kombineerimine ja põimimine

Kombineerimise peamine kriteerium on see, et kõikide helilõikude .srt-failid (nii originaal- kui ka tõlgitud keeles) kombineeritakse korrektselt üheks terviklikuks .srt-failiks, kusjuures ajakoodid on õigesti ümber arvutatud vastavalt lõigu algusajale videos. Vigadeks võivad olla valed ajastused või puuduvad subtiitrid.

Programm saab kombineerimisega väga hästi hakkama ja vigu ei esine. Näiteks *audio_chunk_1_translated.srt* (0-120s) ja *audio_chunk_2_translated.srt* (0-120s, aga video ajas 120-240s) sisu liidetakse kokku, kusjuures teise faili ajakoodidele liidetakse 120 sekundit.

Põimimise Kriteeriumiks on see, et tõlgitud .srt-faili subtiitrid kuvatakse videol õigel ajal, õiges kohas ja defineeritud stiiliga. Vigadeks võivad olla sünkroonist väljas subtiitrid, loetamatu tekst või video renderdamise probleemid.

Genereeritud video_koos_subtiitritega.mp4 mängib subtiitreid sünkroonis kõnega. Esineb üksikuid punkte, kus subtiitrid ruttavad ette, aga 2-minutilised lõigud teevad kindlaks, et kui peaks juhtuma selliseid momente, siis subtiitrid korrigeeritakse kindlalt iga 2 minuti tagant tulevates kontrollpunktides.

Arvutusajad (konkreetse testi põhjal):

- Originaalkeelsete SRT-failide kombineerimine (Srt combination original): 0.05 sekundit.
- Tõlgitud SRT-failide kombineerimine (Srt combination translated): 1.03 sekundit.
- Subtiitrite põimimine 02:50:53 pikkusele videole (Subtitle embedding): 3903.53 sekundit (ligikaudu 1 tund, 5 minutit ja 4 sekundit).

3.3 Kogu protsessi väljund ja hinnang

Antud alampeatükis on kättesaadavad programmi tulemuste väljundid ja on välja toodud kvalitatiivne hinnang.

3.3.1 Väljundid

Lõplik originaalkeelne SRT-fail: https://figshare.com/articles/media/Loengute_T_lkimine_ja_Subtiitrite_Genereerimine_-_V_ljundid/29064503?file=54535403

Lõplik tõlgitud SRT-fail eesti keelde: https://figshare.com/articles/media/Loengute_T_lkimine_ja_Subtiitrite_Genereerimine_-_V_ljundid/29064503?file=54535406

Videofail koos põimitud subtiitritega: https://figshare.com/articles/media/Loengute_T_lkimine_ja_Subtiitrite_Genereerimine_-_V_ljundid/29064503?file=54535409

3.3.2 Kvalitatiivne hinnang

Õnnestumised:

- Arvestades seminari pikkust ja mitut esinejat, suutis süsteem genereerida täieliku transkriptsiooni ja tõlke.
- Arvestades, et paljudel esinejatel pole inglise keel emakeel ja esineb täitesõnu, suutis süsteem teha korraliku transkriptsiooni
- Arvestades eesti keele keerulist grammatikat ja väheseid eesti keelseid andmeid võrreldes suuremate ja populaarsemate keeltega, suutis süsteem teha mõistetava tõlke
- Kasutajamugavus Streamliti liidese kaudu on hea.
- Ajakulude ja API kulu detailne logimine annab hea ülevaate ressursikasutusest.
- Tegemist on odavama ja kiirema lahendusega kui inimtõlk

Ebaõnnestumised ja kitsaskohad:

- Eestikeelsed nimed ja kohad võivad väheste andmete tõttu ja inglise keelt tuvastava transkriptsiooni mudeli tõttu ebakorrektsed olla.
- Erialane terminoloogia seminaril: Andmeteaduse spetsiifilised terminid võivad vajada täiendavat tähelepanu tõlke promptis, et tagada kõige täpsemad vasted sihtkeeles.
- Käänded eesti grammatikas võivad valesti esineda.
- Ajastus ja segmenteerimine: Pikemate monoloogide või kiirema kõne puhul võib subtiitrite segmenteerimine ja ajastus vajada manuaalset korrigeerimist parema loetavuse saavutamiseks.
- Ressursinõudlikkus: Ligi 3-tunnise video töötlemine võttis antud riistvaral kokku ligi 4 tundi, millest suur osa kulus transkribeerimisele. See näitab, et väga pikkade videote puhul on protsess ajamahukas. Võimsam GPU kiirendaks oluliselt transkribeerimist.
- ImageMagick sõltuvus: Subtiitrite põimimisel vajab MoviePy ImageMagick teeki. Selle puudumine või vale seadistus võib põimimisel vigu tekitada või subtiitrite põimimist mitte võimaldada.

3.3.3 Kogu protsessi arvutusaeg

Testitud 02:50:53 pikkuse andmeteaduse seminari töötlemisel olid summeeritud ajakulud järgmised:

- OpenAI kliendi initsialiseerimine: 0.26 sekundit
- Whisper mudeli laadimine: 10.65 sekundit
- Audio eraldamine: 46.85 sekundit
- Audio tükeldamine: 53.46 sekundit
- Transkribeerimine (kokku 86 lõiku): 9207.12 sekundit (≈ 2h 33m 27s)
- Tõlkimine (kokku 86 lõiku): 753.53 sekundit (≈ 0h 12m 34s)
- SRT kombineerimine (originaal): 0.05 sekundit
- SRT kombineerimine (tõlgitud): 1.03 sekundit
- Subtiitrite põimimine: 3903.53 sekundit (≈ 1h 5m 4s)
- Kogu töötlemisaeg (Total processing time): 13977.86 sekundit (ligikaudu 3 tundi, 52 minutit ja 58 sekundit)
- Kogu lõime tööaeg (Thread execution time, sisaldab ka puhastust): 13978.00 sekundit

3.4 Tulemuste lisad

Antud peatükis on lingid failidele, mida kasutati transkriptsiooni ja tõlke hindamiseks. Saadaval on ka näidisvideo programmi kasutamisest.

3.4.1 Transkriptsiooni hindamiseks kasutatud failid

Audio_chunk_7.srt, millest kalkuleeriti WER (Word Error Rate): https://figshare.com/articles/media/Loengute_T_lkimine_ja_Subtiitrite_Genereerimine_-_V_ljundid/29064503?file=54535640

Referentstranskriptsioon *audio_chunk_7.srt-le*: https://figshare.com/articles/media/Loengute_T_lkimine_ja_Subtiitrite_Genereerimine_-_V_ljundid/29064503?file=54535616

3.4.2 Tõlke hindamiseks kasutatud failid

Audio_chunk_25_original.srt: https://figshare.com/articles/media/Loengute_T_lkimine_ja_Subtiitrite_Genereerimine_-_V_ljundid/29064503?file=54535658

Audio_chunk_25_translated.srt: https://figshare.com/articles/media/Loengute_T_lkimine_ja_Subtiitrite_Genereerimine_-_V_ljundid/29064503?file=54535691

3.4.3 Näidisvideo programmi kasutamisest

Näidisvideo: https://figshare.com/articles/media/Loengute_T_lkimine_ja_Subtiitrite_Genereerimine_-_V_ljundid/29064503?file=54547919

3.5 Võimalikud edasiarendused

Kuigi käesolev rakendus pakub funktsionaalset lahendust automaatseks transkribeerimiseks ja tõlkimiseks, tuvastati potentsiaalseid edasiarendusvõimalusi, mis võiksid selle kasutusväärtust suurendada. Nende realiseerimine jäi bakalaureusetöö ajaliste piirangute tõttu käesoleva töö raamest välja.

Üks praktiline ja otsest kasu toov edasiarendus oleks kasutajaliidese täiendamine funktsiooniga, mis võimaldaks transkribeeritud originaalteksti (inglise keeles) ja genereeritud eestikeelseid subtiitreid kõrvuti kuvada ning vajadusel otse redigeerida. See annaks kasutajale võimaluse kiiresti üle vaadata ja parandada vigu, mis automaatses protsessis paratamatult tekivad – näiteks valesti tuvastatud pärisnimesid, erialatermineid või muid tõlke ebatäpsusi. Kasutaja saaks läbi kerida subtiitrifailid, tuvastada probleemkohad ja teha vajalikud parandused.

Kokkuvõte

Tartu Ülikooli Arvutiteaduse Instituudis on märkimisväärne osa õppematerjalidest ja avalikest loengutest, kättesaadavad vaid inglise keeles. See seab piirangud nende materjalide ligipääsetavusele eestikeelsele sihtrühmale, kes ei pruugi inglise keelt piisaval tasemel vallata. Käesoleva bakalaureusetöö peamiseks eesmärgiks oli välja töötada tarkvaraline lahendus, mis automatiseerib ingliskeelsete akadeemiliste loengute transkribeerimise, eesti keelde tõlkimise ning ajastatud subtiitrite genereerimise protsessi, vähendades seeläbi vajadust aeganõudva manuaalse töö järele.

Selle eesmärgi saavutamiseks arendati välja rakendus, mis integreerib kaasaegseid tehisintellekti tehnoloogiaid. Rakendus kasutab OpenAI Whisper mudelit automaatseks kõnetuvastuseks (transkribeerimiseks) ning OpenAI ChatGPT API-t genereeritud transkriptsioonide masintõlkeks inglise keelest eesti keelde. Lisaks rakendati moviepy teeki video- ja helifailide töötlemiseks ning Streamlit raamistikku interaktiivse kasutajaliidese loomiseks.

Rakenduse põhifunktsionaalsus hõlmab kolme järjestikust etappi: esiteks loengu heliraja transkribeerimine tekstiks, teiseks saadud ingliskeelse transkriptsiooni tõlkimine eesti keelde ning kolmandaks tõlgitud teksti põhjal täpselt ajastatud subtiitrite genereerimine levinud SubRip (.srt) vormingus. Välja töötatud lahendus pakub praktilist tööriista ingliskeelse akadeemilise sisu kättesaadavuse parandamiseks Eestis.

Viidatud kirjandus

[1] Civan, A. & Coşkun A. (2016). The effect of the medium of instruction language on the academic success of university students. Educational Sciences: Theory & Practice, 16, 1981–2004. https://files.eric.ed.gov/fulltext/EJ1130741.pdf (15.05.2025).

[2] OpenAI. Whisper. 2022. https://openai.com/index/whisper/ (05.12.2024).

[3] Radford A., Kim J. W., Xu T., Brockman G., McLeavey C., Sutskever I. Robust Speech Recognition via Large-Scale Weak Supervision. arXiv, 2022, arXiv:2212.04356. https://arxiv.org/pdf/2212.04356 (05.12.2024).

[4] OpenAI. Introducing APIs for GPT-3.5 Turbo and Whisper. 2023. https://openai.com/index/introducing-chatgpt-and-whisper-apis/ (15.12.2024).

[5] OpenAI. GPT-4o mini: advancing cost-efficient intelligence. 2024. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/ (15.12.2024).

[6] Streamlit. https://streamlit.io/ (20.04.2025).

[7] Rabiner L., Juang B. Fundamentals of Speech Recognition. Englewood Cliffs, NJ: Prentice Hall PTR. 1993.

[8] Hinton G., Deng L., Yu D., Dahl G., Mohamed A.-R., Jaitly N., Senior A., Vanhoucke V., Nguyen P., Sainath T., Kingsbury B. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. IEEE Signal Processing Magazine, 2012, Vol. 29, No. 6, pp. 82-97. https://ieeexplore.ieee.org/document/6296526 (17.04.2025).

[9] Graves A., Fernández S., Gomez F., Schmidhuber J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. Proceedings of the 23rd international conference on Machine learning, 2006, pp. 369-376. https://www.cs.toronto.edu/~graves/icml_2006.pdf (17.04.2025).

[10] Chan W., Jaitly N., Le Q. V., Vinyals O. Listen, Attend and Spell. arXiv, 2015, arXiv:1508.01211. https://arxiv.org/abs/1508.01211 (17.04.2025).

[11] Koehn P., Och F. J., Marcu D. Statistical Phrase-Based Translation. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, 2003, pp. 48-54. https://www.researchgate.net/publication/220816913_Statistical_Phrase-Based_Translation (17.04.2025).

[12] Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv, 2014, arXiv:1409.0473. https://arxiv.org/abs/1409.0473 (17.04.2025).

[13] Sutskever I., Vinyals O., Le Q. V. Sequence to Sequence Learning with Neural Networks. arXiv, 2014, arXiv:1409.3215. https://arxiv.org/abs/1409.3215 (17.04.2025).

[14] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I. Attention Is All You Need. arXiv, 2017, arXiv:1706.03762. https://arxiv.org/abs/1706.03762 (17.04.2025).

[15] Tättar A. Multilingual machine translation for under-resourced languages. TÜ arvutiteaduse instituudi doktoritöö, 2025 https://dspace.ut.ee/items/bf25ee4e-074a-4c3f-8cec-565b39a3c818 (15.05.2025)

[16] W3C. Web Content Accessibility Guidelines (WCAG) https://www.w3.org/TR/WCAG21/#captions-prerecorded. (15.05.2025).

[17] Krukowski I. SRT files and all you need to know about SubRip subtitles: Lokalise, 2024 https://docs.lokalise.com/en/articles/5365539-srt-files-and-all-you-need-to-know-about-subrip-subtitles (15.05.2025).

[18] W3C. WebVTT: The Web Video Text Tracks Format. 2019. https://www.w3.org/TR/webvtt1/. (15.05.2025).

[19] Clevercast. BBC Subtitle Guidelines. https://www.clevercast.com/bbc-subtitling-guidelines/ (15.05.2025)

[20] Striuk A. M. & Hordiienko V. V. Research and development of a subtitle management system using artificial intelligence,CS&SE@SW 2024: 7th Workshop for Young Scientists in Computer Science & Software Engineering, December 27, 2024, Kryvyi Rih, Ukraine, https://ceur-ws.org/Vol-3917/paper61.pdf (15.05.2025)

[21] Speech and Tech. Forced Alignment. https://www.speechandtech.eu/tech-tools/forced-alignment (15.05.2025)

[22] OpenAI (2024). ChatGPT (GPT-4 Turbo). https://chatgpt.com/.

[23] Word Error Rate (WER) Score, Deepcheeks Glossary

https://www.deepchecks.com/glossary/word-error-rate-wer-score/ (15.05.2025)

[24] OpenAI. Pricing https://openai.com/api/pricing/ (15.05.2025).

Litsents

Mina, Hendrik Püss ,
(autori nimi)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Loengute Tõlkimine ja Subtiitrite Genereerimine
(lõputöö pealkiri)

mille juhendaja(d) on Ardi Tampuu ,
(juhendaja nimi)

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;

2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commonsi litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni;

3. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile;

4. kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Hendrik Püss

15.05.2025