Tallinna Ülikool

Digitehnoloogiate instituut

Informaatika

Vabavaraliste kõnetuvastuse lahenduste võrdlus

Seminaritöö

Autor: Henrik Romanenkov

Juhendaja: Andrus Rinde

Tallinn 2017

Autorideklaratsioon

Deklareerin, et käesolev seminaritöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

....

(kuupäev) (autor)

Sisukord

Sissejuhatus

Kõnetuvastus on kiirelt arenev valdkond, olles kasutusel autodes ja telefonides kiiremateks otsinguteks ning suhtlemiseks, kuid ka erivajadusetega inimeste jaoks ainukeseks lahenduseks suhtlemisel arvutiga. Laialdaselt on see tehnoloogia olnud kasutusel viimased paarkümmend aastat.

Kuigi hetkel on kõige populaarsem puutetundlikust kasutavad tehnoloogia, nähakse tulevikus kõnetuvastuse esiletõusu. Inimene on mõeldud suhtlema, seega milleks on meil vaja vajutada mitmele nupule, kui seda kõike oleks võimalik teha vaid suud liigutades. Siiski võtab üleminek aega ning hetkel ei ole veel ka piisavalt terviklikke lahendusi, et see massidesse jõuaks. Populaarseimad rakendused antud valdkonnas on tänapäeval mobiiltelefonide operatsioonisüsteemides olevad kõnetuvastust kasutavad abivahendid, mis aitavad leida vajaminevat informatsiooni.

Teema valiti seetõttu, et antud valdkond on kiirelt kasvav, kuid hetkel veel mitte nii laialdaselt kasutuses. Lisaks on soov antud seminaritööd kasutada ettevalmistusena bakalaurusetööks, mille raames oleks eesmärgiks valmistada töötav kõnetuvastust kasutav rakendus.

Käesoleva seminaritöö eesmärgiks on anda ülevaade erinevatest vabavaralistest kõnetuvastuse programmidest, kirjeldada, mis on sellise tehnoloogia plussid ja miinused ning milline on erinevus programmide vahel.

Töö eesmärgi saavutamiseks antakse kõigepealt ülevaade kõnetuvastuse ajaloost ning selle taga olevast tehnoloogiast. Töö teises peatükis võrreldakse erinevaid selle valdkonna programme ning süsteeme. Lisaks antakse ülevaade, millistes valdkondades on kõnetuvastus juba kasutusel ning kuidas see aitab ja mõjutab inimesi.

Antud seminaritöö eeldab mõningaid eelteadmisi informaatikas, et arusaada erinevatest terminitest ja seostest. Siiski saavad tööst kasu kõik, kellel on huvi antud valdkonna arengu kohta ning soovivad saada lisateadmisi kõnetuvastusprogrammide tööst.

1. Kõnetuvastus

Kõnetuvastus (inglise *speech recognition*) on inimkõne sisu automaatne äratundmine arvutitehnika poolt (Mereste, 2003). Lisaks kõnetuvastusele kasutatakse sama valdkonna kohta ka mõistet „häältuvastus“. Kas see on ka üks ja see sama? Võrreldes eestikeelseid sõnu „kõne“ ja „hääl“, on arusaadav, et need ei tähenda ühte ja sama asja. Kuidas on aga mainitud terminite tähendusega antud töös uuritavas valdkonnas?

Veebipõhise teatmiku Vallaste andmetel on inglisekeelsed terminid „*speech recognition“* kuid ka „*voice recognnition*“ eesti keeles sama vastega – kõnetuvastus. Nende selgitused on natukene erinevad, kuid „*voice recognition“* selgitusse on kirjutatud, et sama terminit kasutatakse ka kõne tekstiks muundavate programmide kohta.

Selle tulemusena kasutatakse antud töös mõistet „kõnetuvastus“, kuigi on internetis leitav palju materjali, milles räägitakse samast valdkonnast, kasutades mõistet „häältuvastus“.

1.1 Ajalugu

Esimesed märkmed kõnetuvastusest pärinevad juba 18. sajandist, kuid kiirem areng algas 1950ndatel. Tänapäeval on see laialdaselt kasutusel ning on kiiresti arenev valdkond, kuid siiski ei kasuta paljud inimesed seda võimalust.

1.1.1 Maailmas

Inimesed on proovinud luua masinat, mis oskaks rääkida juba 18. sajandi teisest poolest, kuid siis polnud eesmärgiks mitte luua kõne tuvastav ja arusaav, vaid lihtsalt rääkiv masin. Esimene automaatne kõnetuvastusemasin valmistati aastal 1952, mil kolm Bell Labs'i teadlast ehitasid ühest kõnelejast koosneva süsteemi numbrite tuvastamiseks. Nende süsteem töötas otsides iga ütluse võimsuse spektris formandit. 1950ndatel piirdus tehnoloogia ühe kõneleja süsteemiga, mille sõnastik oli umbes kümme sõna. Kahjuks lõpetati 1969. a mitmeks aastaks Bell Labs'is kõnetuvastuse rahastamine, kui mõjukas John Pierce kirjutas avaliku kirja, milles ta oli kriitiline kõnetuvastust arendava teadus vastu, väites, et liiga palju raha on kulutatud süsteemidele, mis on väga algelised ega ole suutelised paranema. (Rabiner & Juang, kuupäev puudub)

Raj Reddy oli esimene inimene, kes hakkas 1960ndate lõpul töötama pideva kõnetuvastuse (continuous speech recognition) kallal. Eelnevad süsteemid nõudsid kasutajalt pausi tegemist pärast igat sõna. Reddy poolt kavandatud kestva kõnetuvastuse süsteemi eesmärgiks oli anda häälkäsklusi malemängu käikude jaoks. (Rabiner & Juang, kuupäev puudub)

Umbes samal ajal leiutasid Nõukogude teadlased dünaamilise ajadeformatsiooni(dynamic time warping, DTW) algoritmi ja kasutasid seda, et luua tuvastaja, mis oli võimeline töötama 200 sõnast koosneva sõnavaraga. DTW algoritm töötles kõnesignaali, jagades selle lühikesteks kaadriteks (nt 10 ms segmentideks) ning töödeldes igat kaadrit kui omaette üksust. Kuigi DTW asendati hiljem uuemate algoritmidega, siis signaali jagamise tehnika kaadriteks jätkus. Kõnelejast sõltumatuse saavutamine oli sel ajal teadlaste peamine probleem. (Wikipedia, kuupäev puudub)

1971. a rahastas DARPA (The Defense Advanced Research Projects Agency) viie aasta vältel kõnetuvastuse uuringuid läbi oma kõnest arusaamise programmi, mille lõppeesmärkide hulka kuulus ka tarkvara, mille minimaalne sõnavara suurus on 1000 sõna. Valitsuse rahastamine taaselustas kõne tuvastamise uurimise, mis oli USA’s suures osas pärast John Pierce'i kirja hüljatud. (Wikipedia, kuupäev puudub)

Hilistel 1960ndatel arendas Leonard Baum Markovi ahela jaoks matemaatika. CMUs (Carnegie Mellon Ülikoolis), hakkasid Raj Reddy õpilased James Baker ja Janet Baker kõnetuvastuse jaoks kasutama Markovi peitmudelit (Hidden Markov Model, HMM). HMM’i kasutamine võimaldas arendajatel siduda erinevaid teabeallikaid, nagu akustika, keel ja süntaks, ühtses tõenäosuslikus mudelis. (Wikipedia, kuupäev puudub)

Fred Jelinek’i juhtimise all lõi IBM häälega aktiveeritava kirjutusmasina Tangora, mis suutis 1980ndate keskpaigaks käsitleda 20000 sõnalist sõnavara. Jelinek’i statistiline lähenemine pani vähem rõhku sellele, kuidas inimese aju töötab ja kõnet mõistab, eelistades kasutada statistilise modelleerimise tehnikaid nagu HMM. Jelinek’i grupp avastas iseseisvalt mooduse, kuidas HMM’e kõneks saada. See oli vastuoluline keeleteadlastega, kuna nende arvates olid HMM’id liiga lihtsustatud, et võtta arvesse mitmeid inimkeelte põhiomadusi. Siiski osutus HMM väga kasulikuks viisiks kõne modelleerimises ning asendas DTW, saades 1980ndatel domineerivaks kõnetuvastuse algoritmiks. 1980ndad nägid ka n-gramm keelemudeli kasutusele võtmist. Samal ajal oli CSELT kasutamas HMM’e, et tuvastada näiteks itaalia keelt. (Wikipedia, kuupäev puudub)

Suur osa selle valdkonna edusammudest võlgnetakse arvutite kiirele arengule. DARPA programmi lõpus aastal 1976 oli parim uurijatele kättesaadav arvuti PDP-10, millel oli 4 MB RAMi. Neid arvuteid kasutades võis 30 sekundi pikkuse kõne dekodeerimine võtta aega kuni 100 minutit. Paar aastakümmet hiljem oli teadlastel juurdepääs tuhandeid kordi suuremale arvutusvõimsusele. Tehnoloogia arenedes muutusid arvutid kiiremaks ning teadlased hakkasid tegelema keerulisemate probleemidega nagu suuremad sõnavarad, kõnelejast sõltumatus, mürakeskkond ja vestluskõne. (Wikipedia, kuupäev puudub)

Alates 1980ndatest aastatest on üleminek keerulisematele süsteemidele iseloomustanud eelkõige ka DARPA rahastamist kõnetuvastusse. Näiteks, sõna veamäära edasine vähenemine toimus selle tõttu, et teadlased arendasid akustilisi mudeleid olema diskrimineerivad, selle asemel, et kasutada maksimaalse tõenäosusega mudeleid. (Wikipedia, kuupäev puudub)

1990ndad nägid esmakordselt kaubanduslikult edukate kõnetuvastustehnoloogiate kasutuselevõttu. Kaks varaseimat toodet olid Voiceworks – 1987. aastal Kurzweil Applied Intelligence'i poolt välja antud kõnetuvastus programm, millel oli 5000 sõnaline sõnavara, kuid sama suur oli ka selle hind, makstes 5000 dollarit ning Dragon Dictate 30K – kõne põhjal töötav kirjutusmasin, mis anti välja 1990. aastal ja mis maksis algselt 9000 dollarit. AT&T võttis 1992. aastal kasutusse häältuvastamisel kõnede töötlemise teenuse, et suunata telefonikõnesid inimoperaatorita. Selleks ajaks oli tüüpilise kommertsliku kõnetuvastamissüsteemi sõnavara suurem kui inimese keskmine sõnavara, mis on umbes 5000 sõna. (Wikipedia, kuupäev puudub)

Raj Reddy endine õpilane Xuedong Huang töötas CMU-s välja Sphinx-II süsteemi. Sphinx-II süsteem oli esimene, mis tegi kõnelejast sõltumatut ja suure sõnavaraga kestvat kõnetuvastamist ning millel oli parim tulemus DARPA poolt 1992. aastal läbi viidud hindamises. Kestva kõne käsitlemine koos suure sõnavaraga oli kõnetuvastamise ajaloos oluline verstapost. Huang lõi 1993. aastal Microsofti kõnetuvastusrühma. Raj Reddy üliõpilane Kai-Fu Lee liitus Apple'iga, kus 1992. a aitas ta välja töötada kõneliidese prototüübi Apple’i arvutitele, tuntud kui Casper. (Wikipedia, kuupäev puudub)

Belgias asuv kõnetuvastuse firma Lernout & Hauspie omandas paljusi ettevõtteid, nende hulgas 1997. a Kurzweil Applied Intelligence ja 2000. a Dragon Systems. L&H kõnetehnoloogiat kasutati Windows XP operatsioonisüsteemis. L&H kõne tehnoloogia ostis ScanSoft, millest sai 2005. a Nuance. Apple litsentseeris algselt Nuance'i tarkvara, et pakkuda kõnetuvastusvõimet oma digitaalsele abistajale Siri. (Wikipedia, kuupäev puudub)

Google'i esimene jõupingutus kõnetuvastuses tuli 2007. aastal pärast Nuance'i teadlaste palkamist. Esimene toode oli GOOG-411, telefonipõhine kataloogiteenus. GOOG-411 salvestised andsid väärtuslikke andmeid, mis aitasid Google'il oma tuvastussüsteeme täiustada. Google'i häälotsingut toetatakse nüüd enam kui 30 keeles. (Google, 2010)

2000ndate aastate algul olid kõnetuvastuses ikka veel domineerivad traditsioonilised lähenemisviisid. Tänapäeval on paljud kõnetuvastuse aspektid üle võetud sügava õppe (inglise *deep learning*) meetodist nimega "pikk lühiajaline mälu (LSTM)". Umbes 2007. aastal hakkas LSTM, mida koolitas ühendusaja ajaline klassifikatsioon (Connectionist Temporal Classification, CTC), osutuma paremaks traditsioonilistest kõnetuvastustest teatud rakenduste puhul. Näiteks koges Google'i kõnetuvastus 2015. aastal märkimisväärset 49%-list jõudluse hüpet CTC-ga koolitatud LSTM-i kaudu, mis nüüd on kõigile nutitelefoni kasutajatele saadaval Google Voice'i kaudu. (Wikipedia, kuupäev puudub)

1.1.2 Eestis

Tallinna Tehnikaülikooli (TTÜ) Küberneetika Instituudi foneetika- ja kõnetehnoloogia laboris tehti esimesed katsetused eestikeelse kõnetuvastusega juba kaheksakümnendate lõpus. Aktiivsemalt hakati kõnetuvastusegatuvastusega tegelema alles 2000ndate keskel. Suure tõuke sellele andis kahe suure eestikeelse kõne andmebaasi (BABEL ja Eesti SpeechDat) loomine, mis võimaldasid treenida juba üsna hästi toimivaid akustilisi mudeleid. Põhiliseks kõnetuvastusega seotud uurimisobjektiks on olnud keelemudel. Eesti keele grammatika, eelkõige keele süntaksist tingitud erinevate sõnavormide rohkus, teeb sõnapõhise laiahaardelise statistilise keelemudeli loomise keeruliseks. (Alumäe, 2011)

2010. a viis TTÜ Küberneetika Instituut läbi projekti „Kõnetuvastus“, mille eesmärgiks oli täiustada juba eksisteerivat kõnetuvastustehnoloogiat ja täiustada olemasolevaid rakendusi ning luua uusi. Projekti tulemusena saadi valmis mitu Android operatsioonisüsteemil põhinevat rakendust, nagu näiteks „Kõneleja“. Kõik need rakendused põhinevad dikteerimisel, kus kõne tehakse tekstiks. Loodi reaalajaline kõnetuvastuse server, mis on mõeldud umbes paarikümne sekundiliste kõnelõikude jaoks. Lisaks paranes aastate vältel erinevates kõnetüüpides esinenud vigade arv. Näiteks, kui konverentsikõnedes oli 2010. a sõnavigade osakaal 37,1%, siis 2014. a oli sama näitaja juba 23,5%. (EKT, kuupäev puudub)

Projektile on ka järg, mille eesmärgiks on sõnavigade osakaalu kõnes veel vähendada ning olemasolevaid rakendusi ja süsteeme kaasajastada. Selles projektis püstitatud sõnavigade eesmärk sai 2017. a tulemuste põhjal täidetud. (EKT, kuupäev puudub)

Lisaks on TTÜs valminud veel kõnesalvestuse brauser, kus saab näha, kui hästi valmistöötatud süsteem erinevat kõne tuvastab. Näha saab nii ühe konverentsi kui ka erinevate raadioprogrammide tõlget. 2011. aastal valmis „Veebipõhine kõnetuvastus“, mille abil saab eestikeelset kõnet sisaldavaid helifaile automaatselt transkribeerida.

Miks ei ole eestikeelsed kõnetuvastus lahendused veel sama kvaliteetsed kui inglisekeelsed lahendused? Vastuseks sellele küsimusele on eesti keele keerukus ning ressursi puudus. Kuigi kõnekorpuse arendamisega tegeles palju inimesi, siis algoritmide ja eesti keele näidete loomisega tegeles TTÜ vanemteadur Tanel Alumäe üki. Eesti keel on ka liialt väike, et sellega saaks tegeletda kommertslikul eesmärgil.

1.2 Kõnetuvastuse tüüpilised kasutusvaldkonnad

- Autosüsteemid

Tavaliselt võimaldab käeline tegevus, näiteks nupu vajutamine roolirattal, aktiveerida kõnetuvastussüsteemi ning sellest antakse juhile teada helisignaaliga. Signaali kostudes on süsteemil nn "kuulamisvahemik", mille käigus aktsepteerib see kõnesisendit. (HyClassProject, kuupäev puudub)

Lihtsaid häälkäsklusi saab kasutada nii telefonikõnede algatamiseks, raadiojaamade valimiseks kui ka muusika esitamiseks nutitelefonist, MP3-mängijast või mälupulgast. Hääletuvastuse võimalused varieeruvad auto margi ja mudeli vahel. Mõned uuemad automudelid pakuvad fikseeritud käskude asemel naturaalset kõnetuvastamist, mis võimaldab juhil kasutada täislauseid ja tavapäraseid väljendeid. Selliste süsteemidega ei ole seega kasutajal vaja mäletada kindlaid häälkäsklusi. (HyClassProject, kuupäev puudub)

- Sõjandus

Viimasel kümnendil on tehtud pingutusi kõnetuvastuse testimiseks ja hindamiseks hävitajate pardal. Erilist tähelepanu on pööranud kõnetuvastuse kasutamisele USA programm F-16 lennukite, Prantsusmaa programm Mirage'i lennukite ja Suurbritannia programm erinevat tüüpi õhusõidukitele. Nendes programmides on hävitajatel edukalt toiminud

kõnetuvastusvahendid, mille hulka kuuluvad raadiosageduste määramine, autopiloodisüsteemi juhtimine, suunamispunkti koordinaatide ja relvade vabastamise parameetrite seadistamine ning lennukuva kontrollimine. (HyClassProject, kuupäev puudub)

Probleemid kõrge tuvastamistäpsuse saavutamiseks stressi ja müra sees esinevad nii helikopterites kui reaktiivlennukites. Akustilise müra probleem on tegelikult helikopterites suurem mitte ainult kõrge müratasemе tõttu, vaid ka sellepärast, et üldiselt ei kanna helikopteri piloot näomaski, mis vähendaks mikrofonis akustilist müra. (HyClassProject, kuupäev puudub)

Viimasel kümnendil on helikopterite kõnetuvastussüsteemide rakendustes tehtud olulisi katse- ja hindamisprojekte. Tulemused on olnud julgustavad ja kõnetuvastusrakendused on sisaldanud järgmist: kommunikatsiooniraadio juhtimine, navigatsioonisüsteemide seadistamine ja automaatse sihtimiskäskude süsteemi juhtimine. (HyClassProject, kuupäev puudub)

Nagu ka hävitajatel kasutatavates rakendustes, on helikopterite häältuvastamise peamine probleem piloodi tõhusus. Selleks, et saavutada seadmetes järjepidevalt jõudluse paranemist, on veel palju teha nii kõnetuvastuses, kuid ka üldises kõnetehnoloogias. (HyClassProject, kuupäev puudub)

- Haridus ja igapäevaelu

Keelte õppimisel saab kõnetuvastus olla kasulik teise keele omandamisel. Lisaks õige hääldamise õpetamisele aitab see ka inimesel arendada rääkimise soravust. Nägemisepuudega õpilased saavad kasutada kõnetuvastustehnoloogiat sõnade edastamiseks ja seejärel kuulata, kuidas arvuti neid esitab. Samuti saavad nad kasutada arvutit, juhtides seda oma häälega, selle asemel, et vaadata ekraani ja vajutada klaviatuuri. (HyClassProject, kuupäev puudub)

Kõne tuvastamine võib lubada õpiraskustega õpilastel saada parematеks kirjutajateks. Üteldes sõnu valjult, saavad nad suurendada oma kirjutamise sujuvust ja leevendada õigekirja, kirjavahemärkide ja muude kirjutamismehhanismide muret. (HyClassProject, kuupäev puudub)

1.3 Kõnetuvastuse komponendid

Kõnetuvastus rakenduse toimimiseks on vajalik läbi viia erinevad töötlusi ning omada mitmesuguseid mudeleid, et kõneleja poolt öeldu saaks süsteemi poole väljundiks kõige täpsema vaste. Sellisteks komponentiteks on eeltöötlus, akustiline mudel, keelemudel ja dekodeerimine. Vastasel juhul ei saa rakendus kõneleja poolt öeldust aru ning arvuti ressursi kasutamine oli asjatu.

1.3.1 Eeltöötlus ja tunnusvektorite arvutamine

Kõnetuvastamise esimeseks sammuks on heli salvestamine ja digitaliseerimine. Pärast seda eeltöödeltakse signaal tuvastuseks sobivale kujule. Signaal lõigatakse lühikesteks lõikudeks, mis on parajasti nii pikad, et selle kestel heli spekter oluliselt ei muutu, ja parajasti nii laiad, et spektrit on võimalik veel arvutada. Pärast seda viiakse läbi analüüs, et leida spektrist olulised tunnused, mis esitatakse reaalarvude vektorina. Saadud tunnused peaksid võimaldama eristada häälikuid, kuid mitte arvestama ebaoluisi signaali aspekte, nagu taustmüra ja kõneleja emotsioonid. (Alumäe, 2011)

1.3.2 Akustilised mudelid

Kõnetuvastussüsteemi üheks oluliseimaks osaks on akustilised mudelid. Akustiliste mudelite abil modelleeritakse häälikuid. Süsteemi hääldussõnastikus on kirjas kõigi rakenduse poolt tuntavatele sõnadele vastavad häälikujadad. Häälikumudelite ja hääldussõnastiku abil saab luua sõnu ja sõnamudelite abil neist lauseid. (Alumäe, 2011)

Akustiliste mudelite põhiomaduseks on võime öelda mis tahes tunnusvektorite jada kohta, kui tõenäoliselt on see signaal just „minu“ poolt esitatud signaal. Selleks kasutatakse akustilistes mudelites normaaljaotustel põhinevaid segumudeleid, mis näitavad, millised on häälikule omased tunnusvektorid ja kuidas nad erinevad. Et modelleerida spektri sõltuvust hääliku asukohast, kasutatakse häälikumudelitena Markovi peitmudeleid. (Alumäe, 2011)

Kõnetuvastuses on tähendatud, et häälikute spekter võib olla tugevas sõltuvuses sellest, millises kontekstis on parajasti häälik. Selle pärast, modelleeritakse häälikuid nn. konktekstist sõltuvate ühikutega: ühel häälikul on palju erinevaid füüsilisi mudeleid, mida kasutatakse automaatselt olenevalt kõne kontekstist. Akustiliste mudelite kõige kasumlikumad

parameetrid arvutatakse suurte kõneandmebaaside põhjal. Kõneandmebaasides on suur hulk erinevate inimeste kõnet koos selgitustega. (Alumäe, 2011)

Selle info põhjal saab akustilisi mudeleid treenida. Saame seada Markovi peitmudelite parameetrid just sellisteks, et need kõneandmebaasiga kõige paremini ühilduksid. Kui andmebaasis on piisav hulk kõnet, peaks treenimise käigus mudelid saama üldistusvõime, ehk sobima ka andmebaasis mitteolevate inimeste kõne tuvastamiseks. (Alumäe, 2011)

1.3.3 Keelemudelid

Akustiliste mudelite abil saame võrrelda süsteemi andmebaasi, kasutaja poolt öelduga. See ei ole aga lause taasesitamiseks piisav. Põhjuseid on siin mitu: inimeste lohakas kõnelemine, sõnalõpud muutuvad dünaamiliselt uue sõna alguseks, sõnade vahel esitatakse kõhklevaid helisi jne. Lisaks on inimeste kõne väga erinev: ühe inimese „o“, häälik, kõlab sarnaselt teise inimese „u“ häälikuga. Isegi, kui häälikude tuvastamine oleks ideaalne, on seda sõnade jadaks keeruline teisendada ilma et teaks midagi keelest, kuna kestva kõne puul puuduvad sõnade vahel pausid, mis aitaksid sõnu piiritleda. (Alumäe, 2011)

Tuvastusmootor suudab tegelikult tuvastada vaid neid sõnu, mis süsteemi keelemudeli sõnastikus esinevad, ning loomuliku keele puhul koosneb sõnastik vaid nendest sõnadest, mis esinevad treeningkorpuses piisavalt tihti. Ainuüksi sõnade loendamine ei lahenda nn homofoonilisuse probleemi: kas häälikujada /kassaoledvalmis/ vastab lausele „Kas sa oled valmis“ või „Kassa oled valmis“? (Alumäe, 2011)

Selle kahetimõistetavuse lahendamiseks tuleb uurida, kuidas sõnu tavaliselt omavahel kombineeritakse. Näiteks ilmneb, et sõnakolmik „kas sa oled“ on üle 100 korra sagedasem, kui sõnapaar „kassa oled“. Selline statistika aitab arvutil leida igale sõnakombinatsioonile õige vaste.Kuna eesti keeles on erinevaid sõnu väga palju kasutatakse keelemudelis sõnade asemel morfeeme. Morfeemid liidetakse pärast tuvastamist uuesti sõnadeks, kuid see protsess toimub kasutajale märkamatult. (Alumäe, 2011)

1.3.4 Dekodeerimine

Dekodeermist nimetatakse protsessi, kus treenitud akustilise mudeli ja keelemudeli ning sõnade häältdussõnastiku abil leiab tuvastusmootor sisendlausele kõige tõenäolisema tuvastuse. Piiratud sõnavaraga tuvastuse puhul on tuvastusprotsess lihtsasti ettekujutatav. Näiteks mobiiltelefonis kasutatava häälvalimisega sarnase rakenduse puhul arvutatakse

sisendsignaali vastavus kõigi telefonis olebate nimede akustilise mudeli suhtes, ning seejärel valitakse neist välja kõige tõenäolisem vaste. (Alumäe, 2011)

Loomuliku ja sidusa kõne tuvastuse puhul on olukord raskem, kuid põhimõte on siiski eelnevaga sarnane: antud juhul tuleb kõigi loomuliku keele lausete hulgast valida välja selline, mis on nii akustilise kui ka keelemudeli seisukohast kõige tõenäolisem. Kuna loomulikus keeles on erinevaid võimalikke lauseid praktiliselt lõpmatult, tuleb otsingus kasutada algoritme, mille abil vähem tõenäolisemad harud sellises lausete puus kohe välistada. Selle tegemata jätmisel võtaks dekodeerimine lause pikkusega võrreldes palju rohkem aega. (Alumäe, 2011)

Sellist otsinguruumi piiramist nimetatakse kärpimiseks (ingl pruning) ning kärpimise ulatuse abil saab dekodeerimise kiirust parandada. Kahjuks võib aga juhtuda, et liigse kärpimise korral lõigatakse ära ka mõningad sellised harud, mis hiljem võivad osutuda vajalikeks ning seetõttu ei tohi seda liiga innukalt rakendada. (Alumäe, 2011)

1.4 Erinevad mudelid, meetodid ja algoritmid

Nii akustiline kui ka keeleline modelleerimine on kaasaegsete statistikapõhiste kõnetuvastusalgoritmide olulised osad. Markovi peitmudeleid on neist kõige populaarsemad, olles kasutuses paljudes süsteemides. Keelelist modelleerimist kasutatakse ka paljude teiste keeltetöötlemise rakenduste puhul, nagu dokumentide liigitamine või statistiline masintõlge.

1.4.1 Markovi peitmudel

Markovi peitmudeleid kasutatakse siis, kui eeldatakse, et kõne on juhuslik protsess. Kaasaegsed üldotstarbelised kõnetuvastussüsteemid põhinevad Markovi peitmudelitel. Need on statistilised mudelid, mis väljastavad sümbolite või koguste jada. HMM’e kasutatakse kõnetuvastuses, kuna kõnesignaali saab vaadelda kui tükeldatud statsionaarset signaali või lühiajalist statsionaarset signaali. Lühikese aja jooksul (nt 10 millisekundit) saab kõnet võrrelda statsionaarse protsessina. Kõnet võib pidada kui Markovi mudelit paljudele juhuslikele otstarvetele. (Paul, 1990)

Näiteks kasutavad seda mudelit populaarseimad kõnetuvastusmootorid Kaldi ja CMUSphinx.

1.4.2 Dünaamilne ajadeformatsioon

DTW on lähenemisviis, mida ajalooliselt kasutatakse kõnetuvastamiseks, kuid mille on nüüd suures osas üle võtnud tulemuslikum HMM. Dünaamiline ajadeformatsioon on algoritm, mis mõõdab sarnasust kahe järjestuse vahel, mis võivad ajas või kiiruses muutuda. DTW-d on rakendatud videole, helile ja graafikale, kuid tegelikult on see võimeline analüüsima kõiki andmeid, mida saab muuta lineaarsele kujule. (Neuro AI, 2013)

Näiteks tuvastatakse sarnasused jalutamisharjumustes isegi siis, kui ühes videos kõndis inimene aeglaselt ja teises käis ta kiirestii või kui toimus kiirendamine ja aeglustamine ühe vaatluse käigus. DTW algoritmi kasutatakse tihti rakendustes, mille esmärgiks on automaatne kõnetuvastamine. (Neuro AI, 2013)

1.4.3 Neurovõrgud

1980ndate aastate lõpus tundus automaatse kõnetuvastuse (automatic speech recognition, ASR) jaoks atraktiivne akustilise modelleerimisega seotud neurovõrk. Sellest ajast alates on neurovõrke kasutatud kõnetuvastuse paljudes aspektides, nagu näiteks nähtuste klassifikatsioonis, isoleeritud sõna tuvastamises, audiovisuaalses kõnetuvastuses ning kõneleja tuvastamises ja kõneleja kohanemises. (Wikipedia, kuupäev puudub)

Erinevalt HMM'idest ei tee närvivõrgud funktsionaalsete statistiliste omaduste kohta eeldusi ja neil on mitu omadust, mis muudavad need atraktiivseks kõnetuvastamise mudeliks. Vaatamata neurovõrkude edukusele lühiajaliste üksuste (nt üksikute telefonide ja isoleeritud sõnade) klassifitseerimisel, on närvivõrgud kestva tuvastamise jaoks harva edukad. Peamiselt seetõttu, et need ei suuda ajutisi sõltuvusi modelleerida. (Wikipedia, kuupäev puudub)

Viimasel ajal on katsetatud mudelitega, mis suudavad tuvastada varjatud ajalisi sõltuvusi ja kasutada seda teavet kõnetuvastuseks. Tänu ebasobivusele luua ajutisi seoseid, saab neurovõrku alternatiivselt kasutada eeltöötluseks HMM'il põhinevate rakenduste jaoks. (Wikipedia, kuupäev puudub)

Sügav neurovõrk (inglise *deep feedforward neural network*, *DNN*) on kunstlik neurovõrk, millel on mitu peidetud kihti sisend- ja väljundkihtide vahel. DNN'id võivad modelleerida komplekssseid mittelineaarseid suhteid. DNN'i arhitektuur koosneb paigutusmudelitest, mis annavad potentsiaali suurema õppimisvõimega ja keerukatel kõneandmetel põhinevate mudelite modelleerimiseks. (Wikipedia, kuupäev puudub)

1.4.4 „End-to-end“ automaatne kõnetuvastus

Alates 2014. aastast on tuntud palju teaduslikku huvi "end-to-end" ASR’i vastu. Traditsiooniline foneetiliselt põhinev (st kõik HMM’il põhinevad mudelid) lähenemine vajab häälduse-, akustika- ja keelemudelite jaoks eraldi komponente ja väljaõpet. End-to-end mudelid õpivad ühiselt kõiki kõnetuvastaja komponente. See on väärtuslik, kuna nii lihtsustakase õppeprotsessi ja kasutuselevõttu. Näiteks on kõigi HMM-põhiste süsteemide jaoks vaja n-grammi keelemudelit ning tüüpiline selline mudel kasutab sageli mitu gigabaiti mälu, mistõttu on need ebapraktilised mobiilseadmetes. Sellest tulenevalt kasutavad tänapäevased ASR süsteemid Google'ilt ning Apple'ilt (alates 2017. a) pilveteenust ja vajavad võrguühendust ega kasuta seadme enda mälu. (Wikipedia, kuupäev puudub)

2. Vabavaraliste kõnetuvastuslahenduste võrdlus

Kõnetuvastusvahendid on kahesuguseid: valmis rakendused ja rakendusliidesed, mille abil töötav rakendus luua. Erinevaid kõnetuvastus süsteeme on palju, kuid kõik ei ole tasuta kättesaadavad. Siiski on ka vabavaralisi rakendusi piisavalt, et viia nende vahel läbi võrdlus. Lisaks erinevatele rakendustele, saab võrrelda ka erinevate operatsioonisüsteemide koosseisus olevaid kõnetuvastuse lahendusi.

Enamusel inimestest on olemas telefon või arvuti. Suurematel tootjatel on oma uusimatesse seadmetesse sisseehitatud kõnetuvastus, mis aitab inimesel seadet kiiremini kasutada. Tehnoloogia on küll sama, kuid erinevatel platvormidel on see erinevalt teostatud.

2.1 Dikteerimine - Mac OS vs Windows OS

Nii Microsoft'il kui ka Apple'il on olemas oma kõnetuvastusprogramm. Kuigi Apple'i oma on pigem dikteerimise rakendus, kuulub see ikkagi kõnetuvastuse kategooriasse. Apple'i seadmetele on sisse ehitatud oma dikteermisrakendus „Apple Dictation". Paljud inimesed teavad, et Apple'i seadmed kasutavad rakendust nimega „Siri", kuid see süsteem on pigem kiireks e-maili saatmiseks või otsingute tegemiseks internetis. Kuigi dikteerimisrakendus töötab Siri baasil, saab seda kasutada ainult tekstitöötluseks. (Esposito, 2017)

Omas vallas on Apple'i rakendus üks parimatest. Sõnavigade arv on väga väike ning süsteem toetab 31 keelt. Rääkida saab järjepidevalt ilma pause tegemata. Lisaks ei pea kasutajal olema internetiühendust, et seda funktsiooni kasutada. (Esposito, 2017)

Windows'i dikteerimine pakub sama, mida Apple. Tekstitöötlust saab teha kiirelt ja mugavalt ilma hiirt ega klaviatuuri kasutamata ning on olemas palju käsklusi, mis on ettevõtte kodulehe pealt leitavad. Vahe Apple'i rakendusega on see, et Windows'i süsteemi kasutades hakkab see õppima kasutaja kõnet tundma. See tähendab seda, et algul on valesti kirjutatud asju palju rohkem kui rivaalil, kuid aja möödudes jääb vigu järjest vähemaks. (Esposito, 2017)

2.2 Mobiilsed abilised - Siri vs Cortana vs Google Assistant

Siri on Apple'i personaalne assistent, mis debuteeris 2011. a, olles esimene seda tüüpi süsteem mobiilses seadmes. Kuigi Siri teeb kõiki vajalikke päringuid, töötab ta kõige paremini üldiste küsimustega. See tuleneb sellest, et Apple on otsustanud klientide privaatsuse esikohale tõsta. Siri suudab siiski teha sissekandeid kalendrisse, avada sõnumeid ja muud sarnast. Väikeseks probleemiks on see, et küsimusi ei saa ise sisse trükkida, kui kõnetuvastus ebaõnnestub - funktsioon, mis on nii Google'i süsteemil kui ka Cortana'l olemas. Lisaks kipub tuvastus tõrkuma, kui üritatakse teha päringuid naturaalselt rääkides. (Nield, 2017)

Google Assistant on tehnikaettevõtte kõnetuvastussüsteem mobiilile. Vaatamata sellele, et antud toode avalikustati alles 2016. a mais, on see populaarse tehisintellekti Google Now edukam edasiarendus. Põhifunktsioonid on samad, kuid ühe suure edasiarendusena on süsteemiga võimalik nüüd ka rääkida. Siiski on antud rakenduse põhiliseks müügiargumendiks süsteemi kooskõla Google'i teiste teenuste ja rakendustega, olgu selleks siis Google Photos või Google Maps. (Nield, 2017)

Windows 10'ga kaasatulev Cortana tehisintellekt on allalaetav nii iOS kui ka Android süsteemile. Kasulik abiline, kui pidevalt kasutatakse mitmeid erineva operatsioonisüsteemiga seadmeid. Tavapärase ilmateate või lähedal asuvate kohvikute otsimisega saab Cortana hästi hakkama, kuigi tema seotus teiste app'idega on vähene. Rakendusel pole küll ühtegi ainulaadset osa ning hetkel on see veel natukene tahuline, kuid süsteem saab väga hästi hakkama ajakava organiseerimisega ning on kättesaadav nii desktop versioonis kui telefonis. (Nield, 2017)

2.3 Kõnetuvastus Chrome bruseris

Chrome toetab päris mitut kõnetuvastusrakendust, mida saad brauserisse lisada. Pea kõik sellised rakendused on dikteerimiseks ehk programm kirjutab teksti vastavalt sellele, mida kõneleja räägib.

Esimene selline rakendus on „Speech Recognition Anywhere". See rakendus, teeb täpselt seda, mida tema nimi eesti keelde tõlkituna tähendab. Omamoodi toimib ta nagu virtuaalne abiline. Kui rakendus on allalaetud, tuleb kasutajal nõustuda mikrofoni kasutamisega ning

pärast seda on see valmis kasutamiseks. Chrome brauseri ülesse äärde ilmub mikrofoni ikoon, mida vajutades aktiveerub laiendus. See programm suudab teha nii otsinguid, kerida lehte ülesse ja alla, vajutada erinevate asjade peale ja palju muud. Hea abiline neile, kes ei saa ise klaviatuuri jaa hiir kasutada. (Baker, 2017)

Järgmiseks rakenduseks on „Speechnotes“, mis on küll mõeldud pigem Android süsteemile, kuid millest on olemas ka brauseris töötav versioon. Tegemist on dikteerimis rakendusega, mis aitab kirjapanna märkmeid. Kuna antud rakendus põhineb Google enda kõnetuvastus mootoritel, siis on sellel väga hea tuvastuse täpsus, olles üle 90%. Seetõttu pole imestatav, et rakendusel on üldiselt head arvustused ning see on kõrgemini hinnatud kõnetuvastuse veebirakendus, sest isegi professionaalsed ning tasulised rakenduse ei suuda alati sellist täpsust tagada. Antud süsteemi plussideks on lihtne kasutus, kuna puudub vajadus allalaadimiseks, lai valik erinevaid keeli (eesti keel puudub) ning väga väike sõnavigade protsent. Miinuseks on teksti kustutamise võimalus puudumine läbi häälkäskluse. See või tekitada suuri probleeme, sest isegi kui rakendus väljastas kõnelja poolt öeldu korrektselt, võib rääkijal tekkida uus mõte või soov väljendada ennast teisit. Selleks puudub tal võimalus häälkäskluse näol ning kasutada tuleb klaviatuuri. (Speechnotes, kuupäev puudub)

Viimaseks rakenduseks antud võrdluses on „SpeechTexter“. Suhteliselt sarnane eelmise rakendusega, aidates muuta kõne tekstiks. Kõnetuvastuseks kasutatakse Google abi, võttes kasutusse nende serverid ja algoritmid. See teeb küll tuvastuse kiireks, kuid samas tähendab see, et annate Google’le võimaluse neid andmeid näha ja kasutada. Rakendus toetab üle 40 keele, mis teeb lihtsaks ka uue keele õppimise, aidates sul kontrollida, kas hääldad sõnu õigesti. Miinuseks antud rakenduse puhul on mobiilse toe puudus. Kuigi rakendus töötab Google Chrome’ga, saab seda kasutada ainult *Desktop* versioonis. Positiivseks aspektiks on rakenduse omadus jätta kirjutatud asjad alles ehk kui oled dikteerimise lõpetanud ja sulged rakenduse, ei kao tekst ära, vaid on uuesti avades olemas, et pooleli jäänud koha pealt kõnelemist jätkata. (SpeechTexter, kuupäev puudub)

2.4 Kõnetuvastus API'd

Kõige poulaarsemad kõnetuvastus API'd (eesti programmiliides) on CMUSphinx, HTK, Julius ning Kaldi. Need neli on põhilised, mida kasutatakse kõnetuvastus programmide loomisel. Kõik neli süsteemi sisaldavad vahendeid efektiivseks kõnetuvastuseks.

2.4.1 CMUSphinx

CMUSphinx on Carnegie Mellon ülikooli poolt arendatav kõnetuvastussüsteem. See koosneb mitmest paketist, mille kasutusvaldkonnad on erinevad. Esimene töötav süsteem valmis 1986. aastal, kasutades HMM'i akustilise mudelina. Süsteemil on seitsme keele tugi ning omadus keelemudelit ise ehitada. Lisaks on Sphinx'i tööriistad disainitud olema sobilikud vähese võimsusega platvormidele. Hetkel on kasutuses kaks versiooni: „Sphinx4" ja „PocketSphinx" ning akustilise mudeli treenija „SphinxTrain" (Carnegie Mellon University, kuupäev puudub)

Esimene neist on mitmekülgse kõnetuvastaja viimane versioon, mis kirjutati täielikult ümber Java programmeerimiskeelde. See muudatus peaks kõnetuvastuse arendamise paindlikumaks tegema. Samas peavad mõned seda muudatust suureks veaks. Näiteks ei toeta Nvidia graafikakaartide paralleelarvutuse platvorm ja programmeerimismudel nimega CUDA ametlikult Javat ning CUDA on vajalik, et treenida neurovõrke, mis on kõnetuvastuse tulevik. Kui tegemist on professionaaliga, pakub Sphinx4 palju võimalusi ning seda võib kasutada ka ärilistel eesmärkidel. (Galvez, 2015)

„PocketSphinx" on tuvastaja, mis on ülesehitatud C keele peale. Tegemist on Sphinx'i versiooniga, mida saab kasutada süsteemidesse sisseehitatuna. See annab sellele versioonile palju kasutus valdkondi. Näiteks saab selle põhjal arendada eraldiseisvaid rakendusi Android operatsioonisüsteemile, olenemata, millise tootja poolt on kasutatav seade toodetud. (Carnegie Mellon University, 2017)

Seda süsteemi kasutavad näiteks eestikeelsed kõnetuvastuse mobiilirakendused „Kõnele ja „Arvutaja".

2.4.2 HTK

HTK on tööriist, mis aitab valmistada HMM'il põhinevaid rakendusi. Kuigi antud tööriista kasutatakse ka muudes valdkondades, keskendume me kõnetuvastusele. HTK koosneb C keelel põhinevatel mudelitel ja teekidel, mis tagavad ka vahendid keele analüüsimiseks,

HMM treenimiseks ning testide läbiviimiseks. Tööriist arendati algselt välja Cambridge ülikooli poolt, kuid 1999. aastal sai HTK litsentis endale Microsoft, kes hiljem andis õigused ülikoolile tagasi, kuid jätkas projekti toetamist. (University Of Cambridge, kuupäev puudub)

HTK sisaldab kõiki vajalikke komponente, et ehitada ise kõnetuvastus mootor ning kuna süsteem on kirjutatud c keeles ja on korrektselt dokumenteeritud, on rakendusi sellest kerge ehitada. HTK suurimaks miinuseks on selle vaba levitamise puudulikkus. Kuigi kood on iseeneset tasuta kättesaadav ning muudetav, ei tohi seda kolmandatele osapooltele edasi anda. Parimaks omaduseks on suur näidete valik, mis teevad kasutamise lihtsamaks. (Thompson, 2017)

2.4.3 Julius

Julius on kõnetuvastus tööriist, mida hakati arendama 1997. aastast ning mis on peamiselt mõeldud kestva kõne tuvastamiseks. Põhiliseks programmeerimis keeleks on C. Headeks külgedeks on vähene mälu kasutus, kiire reaalajas toimuv kõnetuvastus ning väga muudetav vastavalt kliendi vajadusele. (Nagoya Institute of Technology, 2016)

Juliuse suurimaks probleemiks on asjaolu, et see on väljastatud ainult jaapanikeelsete keelemudelitega. Kuigi inglisekeelne mudel on ka olemas, on see väheste võimalustega ning mitte väga hea. Lisaks puudub ka kasutajatugi, vähemalt inglise keelne. (Thompson, 2017)

2.4.4 Kaldi

Kaldi on antud võrdluses kõige uuem kõnetuvastus tarkvara, mille arendus hakkas 2009. Aastal Johns Hopkins'i ülikoolis. Esimene avalik versioon sai valmis 2011. aastal, pärast mida on see saanud kiiresti tuntuks oma kergsuse poolest. Kood on arendatud C++ programmeerimis keeles. Oma olemuselt proovib Kaldi olla sarnane HTK'le, omades paindlikku koodi, mis on kergesti muudetav. (Johns Hopkins University, kuupäev puudub)

Kaldi koodi on võimalik Github'i vahendusel jälgida ja selle on palju panustajaid ning toimub pidev edasiarendus. Kaldi plussideks on tugev kogukond, mis eksisteerib aktiivse foorumi põhjal ning hea kood, mis tagab kiire süsteemi toimimise. Miinuseks on süsteemi puhul kohati keerukas dokumentatsioon. Kuigi vaja minevat informatsiooni on piisavalt, on see tihit liiga keerukas ning mõeldud pigem antud valdkonna spetsialistidele. (Thompson, 2017)

Kaldi on kasutuses TTÜ Küberneetika Instituudi veebipõhises kõnetuvastuses.

3. Kõnetuvastusrakenduste testimine

Antud peatükis viisin läbi erinevate rakenduste testimine. Selgitasin välja, kui hästi saavad progammid kõnetuvastusega hakkama ning millised probleemid esinevad. Et hinnang oleks adekvaatne, kasutasin kõikide lahendusete võrdluseks ühist teksti raamatust „Sirli, Siim ja saladused“, mille autoriks on Andrus Kivirähk. Rakendustest, mis eesti keele tuge ei sisalda, kasutatakse valitud teksti inglisekeelset tõlget.

Võrdluseks valitud tekst: „Asi oli selles, et isa polnud ise ka kunagi varem kalal käinud, aga häbenes sellest pojale rääkida. Ta ei teadnud kalapüüdmisest mõhkugi. Kusagilt oli ta kuulnud, et tarvis läheb ritva, mille otsa pannakse konks ja selle otsa omakorda vihmauss. Aga see oli ka kõik.“

Inglise keelne tõlge: „The thing was that dad had also never gone fishing before and he was ashemed of it. He knew absolutely nothing about fishing. He had heared from somewhere that you need a rod to put a hook onto and an earthworm in turn onto that. But that was all.“

Katset viisin läbi sülearvutisse sisseehitatud mikrofoniga. Olgu siinkohal mainitud, et enamus programme soovitavad kasutada erladiseisvat mikrofoni või mikrofoniga kõrvaklappe.

3.1 Eestikeelsed lahendused

Esimesena võrdlesin dikteerimisrakendust[1], mis teeb reaalajas eesti keele tuvastust. Antud teksti tõlkimisega sai rakendus küllaltki hästi hakkama. Normaalsel kiirusel rääkides polnud ühtegi viga. Suurendades lugemiskiirust, tekkisid väikesed vead. Näiteks „pojale“ asendus sõnaga „pole“. Siiski hindan rakendust kõrgelt ja soovitan seda teistelgi kasutada.

Teisena kasutasin Android süsteemil kasutatavat rakendust „Diktofon“. Sellega saab salvestada helifaili ning siis selle tekstiks muuta. Sellega ilmnes juba natukene probleeme. Kuna rakendus ei tee reaalajas tõlget, siis puuduvad seal ka käsklused kirjavahemärkide jaoks. Väga palju oleneb häältdusest ning pausidest. Pikema pausi kohale tekitas rakendus kohe punkti, kuigi seal oleks pidanud olema mõningal juhul koma. Tekstitõlkega saadi hakkama suhteliselt hästi, kuid vigu esines rohkem võrreldes reaalajalise dikteerimis rakendusega.

[1] https://bark.phon.ioc.ee/dikteeri/index.html

Väga palju sõltub tõlge inimese diktsioonist ning väljendusoskusest. Kõneledes lohakamalt on ka saadud tekst vastav.

Viimasena võrdlesin TTÜ Küberneetika Instituudi foneetika- ja kõnetehnoloogia labori poolt väljaarendatud veebipõhist kõnetuvastust, mis tahab sisendina helifaili, mis saadetakse transkribeeritult sinu emailile. Helifailina kasutasin salvestust, milles lugesin sisse võrdluseks mõeldud teksti. Üldiselt jäin tõlkega rahule, kuid esines üks puuduv punkt ning mõned sõnavead, näiteks oli sõna „vihmauss“ oli asendatud „vihmaussi“. Meeldiv üllatus oli see, et lisaks tõlgitud helifailile, oli emailis veel subtiitrifailid ning sõnade koostamist analüüsivad failid.

3.2 Inglisekeelsed lahendused

Esimseseks veebilehitsejas kasutatavaks rakenduseks oli Speechnotes. Kuigi inglise keel ei ole minu emakeel, pean oma hääldusoskust üpris heaks. Esimest rakendut proovides, lugesin teksti ette kolm korda, iga kord erineva kiirusega, et näha, kas see mõjutab tõlget. Mitte ühelgi korral polnud tekst täiesti õige või paari veaga. Kiiremine lugedes läks tõlge hullemaks. Aeglasemalt lugedes olid vead väikesed. Näiteks sõna „he“ oli asendatud sõnaga „she või „before“ sõnaga „for“. Kiiremini lugedes tekkisid juba väga suured erinevused. Näiteks „Thing was that dad“ muutus fraasiks „Bingo was his dad“. Samas oli sõnu, mis olid kõigis versioonides õiged, nagu „fishing“. Kindlasti aitaks siin vigade vähendamisele kaasa, eraldiseisva mikrofoni kasutamine, mille „kuulmine“ on parem.

Teisena võrdlen veebirakendust SpeechTexter. Jällegi lugesin teksti ette kolm korda, iga kord erineva kiirusega. See rakendus sai palju paremini hakkam kui Speechnotes. Suuri vigu oli väga vähe ja kiiremalt lugedes, ei kasvanud sõnavigade hulk suurel määral. Siiski jäid alles sarnased väikesed vead, kus näiteks „he“ oli tõlgendatud „she“ ning „an“ sõnana „then“.

Viimaste rakendustena proovisin Windows operatsioonisüsteemi sisseehitatud kõnetuvastus programme: Cortana ja Windows Speech Recognition. Varasem kokkupuude kummagi tarkvaraga puudus.

Cortana ülesseadmine on väga lihtne ning kiire. Pärast mõningat kasutust on selge, et Cortana saab enamus kordadest minust väga hästi aru, kuid tema kasutusala jääb minu jaoks piiratuks. Temaga saab avada igasuguseid programme ja teha otsinguid, kuid ta ei suuda teha rakenduste siseseid toiminguid. Näiteks kirjutada tekstiprogrammis sõnu või mängida

Spotify'st muusikat. Windows'i kõnetuvastus, hakkas mulle üha rohkem meeldima. Alguses programmi arusaamine minust oli kohati väga halb. Uurisin selletõttu internetist programmi käsklusi, et kasutust lihtsustada. Pärast materjaliga tutvumist läks tuvastus järjest paremaks. Võrreldes Cortanaga saab seda kasutada ka kolmanda osapoole rakendustes ning tekstitöötluses. Võimalik on teha netis otsingut, kasutada näiteks Facebook'i ning avada operatsioonisüsteemi rakendusi.

3.3 Rakenduste testimise tulemused

Testimise tulemused on väga erinevad ning sõltuvad rakendusest. Eesti keelse rakendusena soovitan kindlasti kasutada TTÜ Küberneetika Instituudi poolt välja töötatud reaalajas tuvastusrakendust. Tõlge oli väga täpne eksides ainult mõnes kohas. Lisaks soovitan kasutada sama asutuse poolt arendatud veebipõhist transkribeerimist, kus saad süsteemile sisestada helifaili, mille tuvastustulemused saad emailile. Üllatavalt saadetakse lisaks tuvastatud kõnest tekkinud tekstifailile ka erinevas formaadis subtiitrifailid ning kõne analüüsi sisaldavad failid.

Inglise keelsete süsteemide hulgas olid tuvastused kehvemad. See võib olla tingitud ka faktist, et kõneleja ei räägi seda keelt emakeelena. Siiski said võrreldud veebilehitsejas töötavatest rakendustes kõnetuvastusega kõige paremini hakkama SpeechTexter. Tuvastatud tekst oli originaalile väga lähedane. Rakendus toetab paljusid keeli ning sobib minu arvates väga hästi dikteerimiseks.

Windowsi süsteemist soovitaksin pigem „Speech Recognition'it. Kuigi selle tuvastus ei olnud Cortana'st parem, on programmi võimalused suuremad. Cortana piirdub ainult ühe käskluse andmisega, millele ei saa teha edasisi pärniguid. Kõnetuvastus lubab avada veebilehti, vajutada erinevatele sõnadele ning teha tööd tekstiredaktorites.

Kokkuvõte

Seminaritöö eesmärgiks oli anda ülevaade erinevatest vabavaralistest kõnetuvastuse programmidest ning neid omavahel võrrelda.

Töö tulemusena sai läbi viidud võrdlused: erinevatel platvormidel olevatest süsteemidest, parimatest süsteemidest veebilehitsejates ning kõnetuvastuse tööriistades, mida kasutatakse reaalsete lahenduste loomises. Selle tulemusena võid leida parima süsteemi, mille põhjal hakata ehitama reaalset kõnetuvastuse rakendust.

Antud töö andis ülevaate kõnetuvastuse ajaloost nii Eestis kui mujal maailmas. Nii saab töös teada, millal hakati kõnetuvastusega tegelema ja kui palju maksid esimesed kommertslikud rakendused. Samuti selgub, milline kõnetuvastuse tarkvara võimaldas esimest korda inimkonnal suheda virtuaalse assistendiga läbi mobiiltelefoni.

Seminaritöös on kirjeldatud komponendid, millest peab kõnetuvastuse rakendus koosnema. Anti ülevaade keele- ja akustilistest mudelitest ning dekodeerimisest, mis on vajalik korrektseks kõne tuvastuseks.

Seminaritööst saab ülevaate erinevatest kõnetuvastuse tehnoloogiatest ja vahenditest. Lisaks on antud töö heaks ettevalmistuseks bakalaureusetööks, andes vajamineva teoreetilise tausta.

Kasutatud kirjanduse põhjal soovitab autor kasutada kõnetuvastuslahenduse loomiseks kas Kandi või CMUSphinx'i kõnetuvastusmootorit, kuna antud süsteemid on ka hetkel edasiarendamisel ning on vabavaraliselt kättesaadavad. Lisaks on mõlemad süsteemid kasutuses eesti keelses kõnetuvastuses.

Kasutatud kirjandus

Alumäe, T. (2011). Allikas: *Kõnetuvastus*. https://phon.ioc.ee/dokuwiki/doku.php?id=konetuvastus.et

Baker, J. (2017). Allikas: Speech Recognition Anywhere - Chrome Extension: http://www.seabreezecomputers.com/speech/

Carnegie Mellon University. (2017). Allikas: Pocketsphinx as standalone app on Android wearables: https://cmusphinx.github.io/2017/03/pocketsphinx-as-standalone-app-on-android-wearables/

Carnegie Mellon University. (kuupäev puudub). Allikas: About CMUSphinx: https://cmusphinx.github.io/wiki/about/

Chitu, A. (2010). Allikas: Google Shuts Down GOOG-411: https://googlesystem.blogspot.com.ee/2010/10/google-shuts-down-goog-411.html#gsc.tab=0

EKT. (kuupäev puudub). *Kõnetuvastus*. Allikas: Eesti Keeletehnoloogia Riiklik Porgramm: https://www.keeletehnoloogia.ee/et/ekt-projektid/konetuvastus

EKT. (kuupäev puudub). *Projektist*. Allikas: Eesti Keeletehnoloogia Riiklik Programm: https://www.keeletehnoloogia.ee/et/ekt-projektid/konetuvastus-2/projektist

Esposito, E. (Juuni 2017. a.). *The Beginner's Guide to Dictation Software: The Best Apps for Voice to Text Productivity*. Allikas: Zapier: https://zapier.com/blog/best-text-dictation-software/

Galvez, D. (2015). Allikas: Which is the best open-source ASR for non-commercial usage? Is HTK still the best, given its long history and rich documents?: https://www.quora.com/Which-is-the-best-open-source-ASR-for-non-commercial-usage-Is-HTK-still-the-best-given-its-long-history-and-rich-documents

Google. (2010). Allikas: Goodbye to an old friend: 1-800-GOOG-411: https://googleblog.blogspot.com.ee/2010/10/goodbye-to-old-friend-1-800-goog-411.html

HyClassProject. (kuupäev puudub). *Design And Implementation Of Voice Recognition System*. Allikas: http://www.hyclassproject.com/design-and-implementation-of-voice-recognition-system.html

Johns Hopkins University. (kuupäev puudub). Allikas: About the Kaldi project: http://kaldi-asr.org/doc/about.html

Mereste, U. (2003). *Majandusleksikon.* Eesti Entsüklopeediakirjastus.

Nagoya Institute of Technology. (2016). Allikas: Julius: Open-Source Large Vocabulary Continuous Speech Recognition Engine: https://github.com/julius-speech/julius

Neuro AI. (2013). *Speech Recognition.* Allikas: http://www.learnartificialneuralnetworks.com/speechrecognition.html

Nield, D. (April 2017. a.). *Siri vs Google Assistant vs Cortana vs Alexa: battle of the AI assistants*. Allikas: T3: https://www.t3.com/news/siri-vs-google-assistant-vs-cortana-vs-alexa-battle-of-the-ai-assistants

Paul, D. (1990). *Speech Recognition Using Hidden Markov Models.* Allikas: Scribd: https://www.scribd.com/document/267210002/Speech-Recognition-Using-Hidden-Markov-Models

Rabiner, L., & Juang, B. (kuupäev puudub). Allikas: Automatic Speech Recognition – A Brief History of the Technology: http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LALI-ASRHistory-final-10-8.pdf

Speechnotes. (kuupäev puudub). Allikas: Speechnotes: https://speechnotes.co/

SpeechTexter. (kuupäev puudub). Allikas: SpeechTexter: https://www.speechtexter.com/about

Thompson, C. (2017). Allikas: Open Source Toolkits for Speech Recognition: https://svds.com/open-source-toolkits-speech-recognition/

University Of Cambridge. (kuupäev puudub). Allikas: What is HTK?: http://htk.eng.cam.ac.uk/

Wikipedia. (kuupäev puudub). Allikas: Speech Recognition: https://en.wikipedia.org/wiki/Speech_recognition