Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Ville Seeste

Puheentunnistus Pepper-robotissa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Insinöörityö

10.5.2019

Tekijä Otsikko Sivumäärä Aika	Ville Seeste Puheentunnistus Pepper-robotissa 33 sivua 10.5.2019
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Smart Systems
Ohjaajat	Lehtori Peter Hjort

Insinöörityössä tutustuttiin puheentunnistusjärjestelmien toimintaperiaatteisiin. Tavoitteena oli testata kolmannen osapuolen tarjoaman puheentunnistuspalvelun käyttöä Pepper-robotissa. Palvelun tulisi pystyä vapaaseen puheentunnistukseen suomen kielellä. Tarkoitus oli myös arvioida, kannattaako palvelua ottaa käyttöön robotin ensisijaiseksi puheentunnistusjärjestelmäksi.

Työssä testattiin Google Cloudin tarjoamia puheentunnistuspalveluita. Kokonaisen äänitiedoston lähettäminen palveluun osoittautui liian hitaaksi vaihtoehdoksi Pepperin keskusteluSovelluksessa. Tämän takia tarkempaan testaukseen valittiin Googlen puheentunnistuksen nopeampi versio, jossa lähetetään äänivirtaa palvelulle.

Palvelua testattaessa havaittiin, että robotin päässä sijaitsevat tuulettimet synnyttävät puheentunnistuksen kannalta haitallista häiriääntä. Ongelmaa lähdettiin korjaamaan vähentämällä haitallisen häiriöäänen määrää äänisignaalissa. Häiriääntä yritettiin vähentää käyttämällä kaistanpäästösuodatinta ja Fourier-analyysiin perustuvaa häiriönpoistoalgoritmia.

Alkuperäisen ja käsitellyn puheen puheentunnistuksesta saatuja vastauksia vertailtiin käyttäen niiden WER-sanavirhetuloksia. Tuloksista selvisi, että kokonaisen äänitiedoston lähettäminen palveluun on tarkempaa kuin puheen lähettäminen äänivirtana. Havaittiin myös, että kaistanpäästösuodattimella ei ollut merkittävää vaikutusta puheentunnistustarkkuuteen. Häiriönpoistoalgoritmin käyttö paransi tarkkuutta äänivirtaa lähetettäessä mutta ei silloin, kun palvelulle lähetettiin kokonainen äänitiedosto.

Tehtyjen testien perusteella arvioitiin, että Googlen tarjoamaa ulkoista puheentunnistusta ei kannata ottaa robotin ensisijaiseen käyttöön. Työstä saatuja tuloksia voidaan hyödyntää Pepper-robotin ulkoisen puheentunnistamisen parannuksissa.

Avainsanat	Pepper, puheentunnistus, DSP, WER

Author Title Number of Pages Date	Ville Seeste Speech recognition on the Pepper robot 33 pages 10 May 2019
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Smart Systems
Instructors	Peter Hjort, Senior Lecturer

The purpose of this thesis was to get familiar with basic operating principles of the speech recognition systems. The main goal was to test a third party speech recognition system for the Pepper robot and evaluate if it should be implemented on the robot. The chosen speech recognition system service should be able to perform free speech recognition and be available in Finnish language.

Google Cloud speech recognition was elected to be tested as a third party speech recognition system. Sending whole audio files to the Google Cloud service proved to be too slow a solution to be used with dialog applications. For this reason, a more profound testing was done on the Google Cloud streaming speech recognition version.

When testing the system, it was found that the fans in the head of the robot are producing harmful background noise, which affected negatively on speech recognition accuracy. Noise reduction for the audio signal was assumed to be a solution for that problem. In order to reduce noise, bandpass filter and noise reduction algorithm was tested on the audio signal.

Original and processed speech was tested on the speech recognition system and results were compared using Word Error Rate (WER). Results showed that sending whole audio files to the service is more accurate than streaming the audio signal. It was also noticed that bandpass filters did not make significant difference on speech recognition accuracy. Noise reduction algorithm was able to improve speech recognition accuracy when streaming the audio to the service but not when sending the whole audio file.

With the results from the test, it can be determined that is it not reasonable to replace Pepper's internal speech recognition system with the Google Cloud speech recognition service. Results from this thesis can be used to further develop the external speech recognition system on the Pepper robot.

Keywords	Pepper, speech recognition, DSP, WER

Sisällys

Lyhenteet

DFT	*Discrete Fourier transform*. Diskreetin ajan Fourier-muunnos, jonka avulla signaali voidaan esittää taajuustasossa.
DNN	*Deep neural network*. Syvä neuroverkko, joka koostuu useasta piilokerroksesta.
FFT	*Fast Fourier transform.* Nopea versio diskreetistä Fourier-muunnoksesta.
GMM	*Gaussian mixture model*. Useista normaalijakaumista koostuva tilastollinen malli.
HMM	*Hidden Markov model*. Kätketty Markov-malli on tilastollinen malli, jota käytetään paljon muun muassa puheentunnistuksessa.
MFCC	*Mel frequency cepstral coefficients*. Puheäänteen piirteiden analysoinnissa käytettävä kerroin.
RNN-T	*Recurrent neural network transducers*. Rekursiivinen neuroverkko, joka pystyy vastaanottamaan sarjana tulevaa syötettä.
STT	*Speech to Text*. Tarkoittaa prosessia, jossa puhe muutetaan sitä vastaavaan tekstimuotoon.
WER	*Word error rate*. Sanavirheiden suhdeluku verrattuna alkuperäiseen tekstiin. Sitä käytetään usein puheentunnistuksen suorituskyvyn testauksessa.

1 Johdanto

Robottien käyttäminen esimerkiksi erilaisissa asiakaspalvelutehtävissä tulee todennäköisesti yleistymään tulevaisuudessa tekniikan kehittymisen myötä. Tämän seurauksena myös puhekäyttöliittymien käytön määrä kasvaa ihmisten arjessa. Puheentunnistuksen on toimittava hyvin, että asiointi robotin kanssa puheen välityksellä olisi sujuvaa ja miellyttävää. Jotta robotin kanssa voitaisiin olla vuorovaikutuksessa puheen välityksellä, on sen yritettävä ymmärtää käyttäjän puhuman puheen tarkoitus. Puheen muuntaminen tekstiksi on kriittinen osa puheen ymmärtämisen prosessia koneen avulla, ja näin ollen sen toimivuus on todella tärkeää esimerkiksi keskustelun kaltaisissa sovelluksissa.

Tämän insinöörityön tarkoitus oli tutustua puheentunnistusjärjestelmien toimintaan ja tästä saadun tiedon avulla tutkia, voidaanko Pepper-robotin puheentunnistusta parantaa ulkoisen puheentunnistuspalvelun avulla. Insinöörityö tehtiin OP:lle, jolla on tämän insinöörityön kirjoittamisen hetkellä käytössään 2 Pepper-robottia. Robotteihin on liitetty OP:n liiketoimintaan liittyviä toiminnollisuuksia, joita ne kiertävät esittelemässä eri pankkikonttoreissa ja tapahtumissa ympäri Suomen.

Robottien mukana tulevassa sisäisessä puheentunnistuksessa on tiettyjä rajoitteita, joten haluttiin tutkia, voidaanko rajoitteita vähentää käyttämällä ulkoista puheentunnistuspalvelua. Parantamalla robotin puheentunnistusta voidaan samalla parantaa keskustelun sujuvuutta robotin kanssa. Tämän seurauksena asiakaskokemus robotin kanssa muuttuu kokonaisuudessaan miellyttävämmäksi asiakkaalle.

Puheen äänisignaalissa esiintyy usein paljon häiriötekijöitä, kuten taustamelua. Äänisignaalissa esiintyvä häiriö vaikeuttaa puheentunnistusta huomattavasti, eikä puheen erottamista muusta äänisignaalista ole vielä saatu kunnolla ratkottua. Robotin tuulettimista aiheutuva häiriö huonontaa nauhoitetun puheen äänisignaalin laatua. Insinöörityössä tutkittiin myös, voidaanko äänisignaalia käsittelemällä parantaa ulkoisesta puheentunnistuspalvelusta saatujen vastauksien tarkkuutta.

2 Puheentunnistus

2.1 Johdanto puheentunnistukseen

Puheentunnistuksella tarkoitetaan teknologiaa, jonka avulla voidaan muuntaa ihmisen puhe sitä vastaavaan tekstimuotoon. Puhe on ihmisen äänihuulten tuottamaa johonkin kieleen perustuvaa puheääntä. Kaikki ihmisen puhumat kielet koostuvat puheäänteistä, joita kutsutaan foneemeiksi. Eri foneemien yhdistelmät muodostavat kielen sanoja. Ihminen muodostaa puhuessaan äänihuulillaan peräkkäisiä äänteitä, jotka muodostuvat sanoiksi ja peräkkäisistä sanoista muodostuu lauseita. Puheääni kulkeutuu ääniaaltoina puheen vastaanottimeen, esimerkiksi kuuntelijan korvaan tai mikrofoniin. (1, s. 31–33; 2, s. 5–6.)

Bell-laboratorion vuonna 1952 kehittämä Andrey-niminen laite oli ensimmäisiä puheentunnistusjärjestelmiä. Andrey oli yksinkertainen puheentunnistusjärjestelmä, ja se kykeni tunnistamaan yksittäisiä numeroita nollasta yhdeksään. Hyvän tunnistustarkkuuden takaamiseksi laiteen asetuksia täytyi virittää puhujan äänen ominaisuuksien mukaan ja puhujan tuli pitää pieni tauko lausuttavien numeroiden välissä. (2, s. 54–61.)

Puheentunnistusteknologia parani huomattavasti 1980-luvulla, kun kätketty Markov-malli (Hidden Markov Model, HMM) otettiin käyttöön foneemien tunnistamisessa. HMM-mallin avulla voidaan arvioida sanojen todennäköisyyksiä foneemien perusteella, ja malli on edelleen yleisesti käytössä puheentunnistusjärjestelmissä. 2010-luvulla syviin neuroverkkoihin perustuvat puheentunnistusjärjestelmät ovat parantaneet puheentunnistusjärjestelmien tarkkuutta ja laajentaneet niiden sanavarastoa huomattavasti. (3, s. 299–308.)

Puheentunnistusta on perinteisesti käytetty esimerkiksi sanelusovelluksissa ja ääniohjauksessa autoissa ja kodin viihdejärjestelmissä. Puheentunnistusteknologian kehittyminen tähän päivään on mahdollistanut uudenlaisten sovelluksien syntymisen. Uutena sovelluksena voidaan pitää esimerkiksi älykaiuttimia, jotka ovat viime aikoina tulleet suosituiksi. Myös mobiililaitteita voidaan hallita puheen avulla. Tästä esimerkkinä on Applen

virtuaaliassistentti Siri, jonka avulla puhelimen käyttäjä voi hakea tietoa verkosta ja hallinnoida laitetta ilman fyysistä kosketusta laitteeseen. Kaikki edellä mainitut sovellukset vaativat puheentunnistusteknologiaa, jotta niitä voisi ohjata puheella.

Puheentunnistusjärjestelmän pääasiallinen tavoite on arvioida todennäköisin sanasarja järjestelmään syötetystä äänisignaalista. Tämä tavoite voidaan mallintaa matemaattisesti kaavaa 1 käyttäen.

$$\widehat{W} = \underset{W}{argmax}\, P(W|O) \quad , W \in \mathcal{L} \tag{1}$$

Kaavan 1 W esittää arvioitavaa sanasarjaa ja O vastaa puheen akustista syötettä. Akustinen syöte tarkoittaa kompaktia piirrevektorijonoa, joka lasketaan puhesignaalista akustisessa esikäsittelyvaiheessa. $\mathcal{L}$ tarkoittaa valitun kielen kaikkia mahdollisia sanasarja -yhdistelmiä. Kaava voidaan muuntaa helpommin laskettavaan muotoon (yhtälö 2) Bayesin teoriaa käyttäen.

$$\widehat{W} = \underset{W}{argmax}\, \frac{P(O|W)\,P(W)}{P(O)} \quad , W \in \mathcal{L} \tag{2}$$

$P(O|W)$ vastaa akustisen syötteen O todennäköisyyttä annetulla W-sanasarjalla. Akustisen syötteen todennäköisyys voidaan laskea foneemimallin avulla. Koska akustinen syöte $P(O)$ on sama jokaiselle sanasarjan W vaihtoehdolle, voidaan yhtälö yksinkertaistaa yhtälön 3 muotoon.

$$\widehat{W} = \underset{W}{argmax}\, P(O|W)\, P(W) \quad , W \in \mathcal{L} \tag{3}$$

$P(W)$ eli sanasarjan todennäköisyys lasketaan käyttämällä ennalta opetettua kielimallia sanojen esiintymisen arvioimiseen lauseessa. Yksinkertaistettuna puheentunnistusjärjestelmästä saatu teksti on sanojen akustisiin ja kielellisiin todennäköisyyksiin perustuva suurin mahdollinen tulos. (4.)

Vaikka puheentunnistus on kehittynyt paljon sitten 50-luvun, kaikkia siihen liittyviä ongelmia ei ole pystytty ratkaisemaan. Vaikeuksia tuottaa esimerkiksi laajan sanaston tunnis-

taminen ja puheentunnistaminen meluisissa olosuhteissa. Puheelle on vaikeaa muodostaa yleisiä tilastollisia malleja, koska puhesignaalin ominaisuuksiin vaikuttaa hyvin moni eri tekijä. Puhesignaalin akustisiin ominaisuuksiin vaikuttaa esimerkiksi taustalla kuuluva melu, mikrofoni, jolla puhetta tallennetaan, ja ympäristö, missä puhe tallennetaan. Taustalla kuuluva ylimääräinen puhe on järjestelmälle hyvin vaikeaa suodattaa, koska se muistuttaa läheisesti tunnistettavaa puhesignaalia. Taustamelun suodattamiseen on vaikea kehittää yleismallillista suodatinta, sillä taustamelun voimakkuus ja piirteet vaihtelevat paljon ympäristöstä ja mikrofonista riippuen. (5.)

Myös puhujan puhetyyliin liittyvät tekijät kuten puhenopeus, aksentti, sanojen ääntäminen ja äänenkorkeus tekevät puheentunnistamisesta monimutkaisen ongelman ratkaistavaksi. Jokaisen puhujan uniikki puhetyyli ja muut puhesignaaliin vaikuttavat tekijät tekevät täydellisen matemaattisen mallin rakentamisesta puheentunnistamiseen hyvin haastavaa. Puheentunnistuksen tarkkuutta voidaan parantaa, kun järjestelmän tilastollisia malleja tehdessä otetaan huomioon puhuja, mikrofoni ja mahdolliset ympäristön aiheuttamat haitalliset äänet. Puheentunnistussovellukset ovat yleensä kuitenkin puhujariippumattomia ja niitä käytetään monilla eri laitteilla, monissa eri ympäristöissä. Näin ollen monissa tapauksissa järjestelmälle ei voida opettaa noin tarkkoja ominaisuuksia puhesignaalista tilastollisia malleja kehitettäessä. (6, s. 341–343.)

Puheentunnistusjärjestelmien tarkkuutta voidaan parantaa ottamalla huomioon puheen kielelliset merkitykset akustisten ominaisuuksien lisäksi. Järjestelmälle voidaan luoda sanastomalli (leksikko), joka määrittää sanastossa olevien sanojen ääntämistavan. Näin voidaan rajata mahdollisia tunnistusvaihtoehtoja kieleen perustuen. Ihmisen puheella on yleensä jokin merkitys ja tilastollisten kielimallien avulla voidaan arvioida peräkkäisten sanojen esiintymistodennäköisyyksiä. Kielimalli on sanojen ja niiden välisten suhteiden tilastollinen todennäköisyysmalli. Kielessä on monia sanoja, jotka kuulostavat melkein identtisiltä, mutta ihminen tunnistaa ne toisistaan puheen kontekstin perusteella. On esimerkiksi todennäköisempää, että puhuja on sanonut: ”Puu kaatuu” kuin: ”Luu kaatuu”. Järjestelmän kannattaa siis ottaa myös sanojen välinen konteksti huomioon, koska sanojen tunnistaminen pelkkien äänteiden perusteella voi joissain tapauksissa olla haastavaa. (6, s. 339.)

Täydellinen puheentunnistusjärjestelmä pystyisi muuntamaan tekstin aina sataprosenttisesti oikein puhujasta, kielestä, puheen laadusta ja muista puhesignaaliin vaikuttavista tekijöistä huolimatta. Nykyaikaiset puheentunnistusjärjestelmät perustuvat tilastollisiin malleihin, jotka pitää etukäteen kouluttaa järjestelmälle. Vaikka tilastollisten mallien opetusaineisto olisi kuinka laaja ja monipuolinen, on lähes mahdotonta päästä täydelliseen tunnistamistulokseen nykyaikaisilla käytössä olevilla menetelmillä. Parhaat nykyiset puheentunnistusjärjestelmät ovat yltäneet noin 95 prosentin tunnistustarkkuuteen ideaaliolosuhteissa. Normaaleissa käyttöolosuhteissa puheentunnistustarkkuus voi kuitenkin laskea huomattavasti ideaaliolosuhteissa tehtyjen testien tuloksista. (7; 8.)

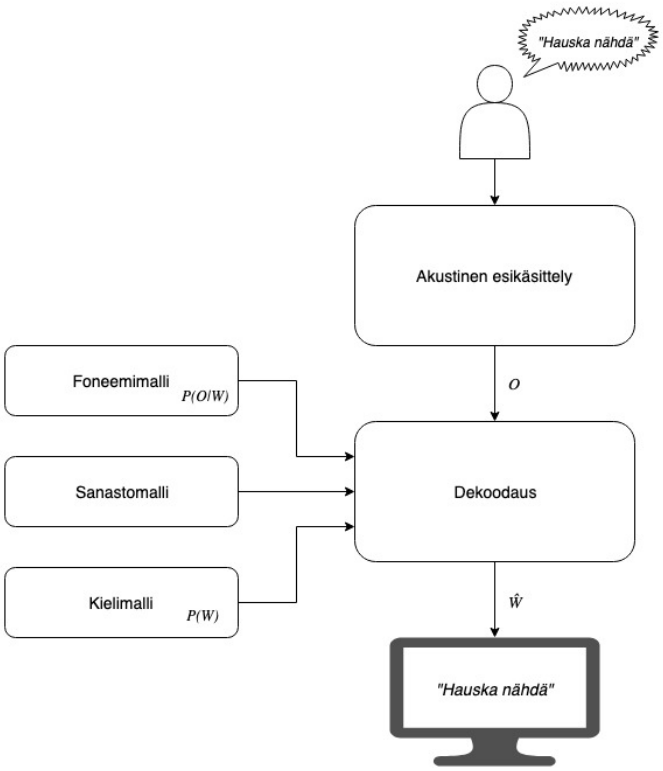
2.2 Puheentunnistamisen vaiheet

Monet nykyiset tuotannossa olevat puheentunnistusjärjestelmät perustuvat äänteiden tunnistamiseen puheäänestä foneemimallin avulla. Tyypillisen äänteiden tunnistamiseen perustuvan puheentunnistusjärjestelmän toimintavaiheet (kuva 1) voidaan jakaa kolmeen päävaiheeseen:

1. Akustinen esikäsittelyvaihe, jossa puhesignaalin digitaalisesta muodosta yritetään poimia puheelle ominaisia piirteitä ja suodattamaan pois epäolennainen informaatio. Puheesta saaduista ominaisuuksista muodostetaan piirrevektoreita, joita käytetään äänteiden analysoinnissa foneemimallissa.

2. Todennäköisyyksien laskeminen tilastollisten mallien avulla. Foneemimalli, sanastomalli ja kielimalli ovat tavallisesti käytettyjä järjestelmän komponentteja.

3. Dekoodausvaihe, jossa järjestelmä hakee jonkin hakualgoritmin avulla malleista saatujen todennäköisyyksien perusteella parhaan mahdollisen tuloksen

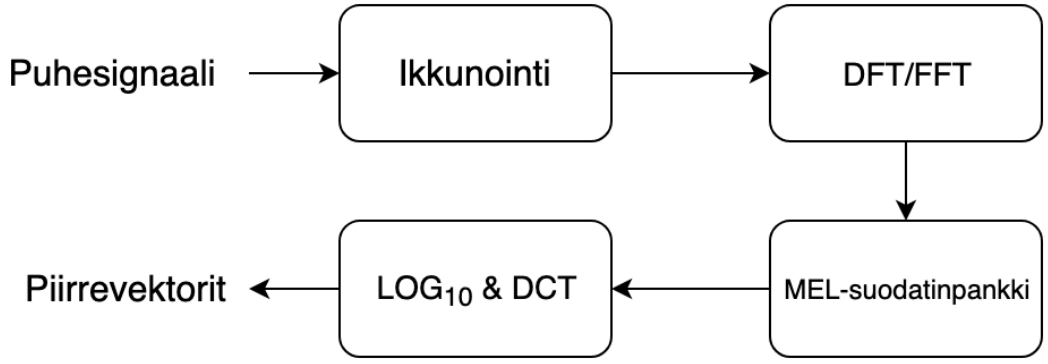


Kuva 1. Kuvaus perinteisen puheentunnistusjärjestelmän rakenteesta.

Akustisessa esikäsittelyvaiheessa äänisignaali muutetaan digitaaliseen muotoon, joka sen jälkeen jaetaan pienempiin, tyypillisesti noin 25 millisekuntia pitkiin osiin 10 millisekunnin välein. Jokaisesta osasta lasketaan taajuusspektri, eli signaalin taajuusjakauma. Taajuusspektristä pyritään ottamaan talteen puheen äänteille ominaisia piirteitä ja suodattamaan pois kaikki epäolennaiset piirteet kuten taustamelu. Taajuusspektrin analysoinnissa keskitytään taajuusalueille, jotka ovat oleellisia puheelle. Taajuusspektrianalyysissä on esimerkiksi turha analysoida 20 kHz:n ylittäviä taajuuksia, koska suurin osa ihmisen puheen informaatiosta on alle 10 kHz:n taajuusalueella (9). Signaalista kerätyistä ominaisuuksista muodostetaan piirrevektori, joka on kompakti versio taajuusspektristä. Piirrevektori sisältää äänteen tunnistamisen kannalta oleellista informaatiota, jota käytetään foneemien tunnistusvaiheessa. Kompakti piirrevektorin koko on tärkeää, koska se nopeuttaa todennäköisyyslaskelmia myöhemmissä vaiheissa. (6, s. 336–337.)

MFCC (Mel frequency cepstral coefficients) (kuva 2) on yleinen tapa piirrevektoreiden muodostamiseen (9). Metodissa digitaalinen äänisignaali jaetaan pienenpiin limittäisiin

osiin. Tämän jälkeen signaalin osille suoritetaan diskreetti Fourier-muunnos (DFT, Discrete Fourier Transform), josta tuloksena saadaan osassa esiintyvien taajuuksien amplitudijakauma. Mel-asteikkoa käytetään seuraavassa vaiheessa, että saadaan laskettua asteikoin suodattimien muodostamien alueiden taajuuksien energiatasojen summat. Mel-asteikko simuloi ihmisen kuulolle ominaisien taajuusalueiden herkkyyttä. Näistä summista lasketaan logaritmi, jonka jälkeen lasketaan niistä diskreetti kosinimuunnos (DCT, Discrete cosine transform). (9.)



Kuva 2. MFCC on yleinen tapa piirrevektoreiden muodostamisessa.

Sanat voidaan yksinkertaisesti mallintaa sarjana foneemeja eli äänteitä. Foneemit ovat puheen pienin rakenteellinen osa, jota muuttamalla voidaan vaihtaa sanojen merkitystä. Esimerkiksi sanan *pää* merkitys muuttuu vaihtamalla sanan ensimmäinen äänne /p/ äänteeseen /s/. Puhuja on tällöin sanonut sanan *sää*. Foneemien erottelua puheesta vaikeuttaa se, että puhe on jatkuvaa äänteestä toiseen siirtymistä, eikä puhe koostu vain peräkkäisistä selkeästi erottuvista äännesegmenteistä (1, s. 87).

Foneemimallin tehtävä puheentunnistusprosessissa on tunnistaa äänisignaalissa esiintyvät foneemit. Foneemimallivaihe on tärkeä osa puheentunnistusjärjestelmässä, joten puheentunnistuksen tarkkuus määrittyy paljon sen mukaan, miten hyvin foneemimalli onnistuu tehtävässään. Foneemimallissa jokaiselle yleiselle äänteelle on laskettu opetusdatan avulla yksi tai useampi tilastollinen malli. Useamman mallin opettaminen joillekin äänteille on tarpeellista, koska ne voivat kuulostaa hyvinkin erilaisilta riippuen siitä, mitkä ovat sen viereiset äänteet. Äänteiden tunnistamisessa yleisin käytetty matemaat-

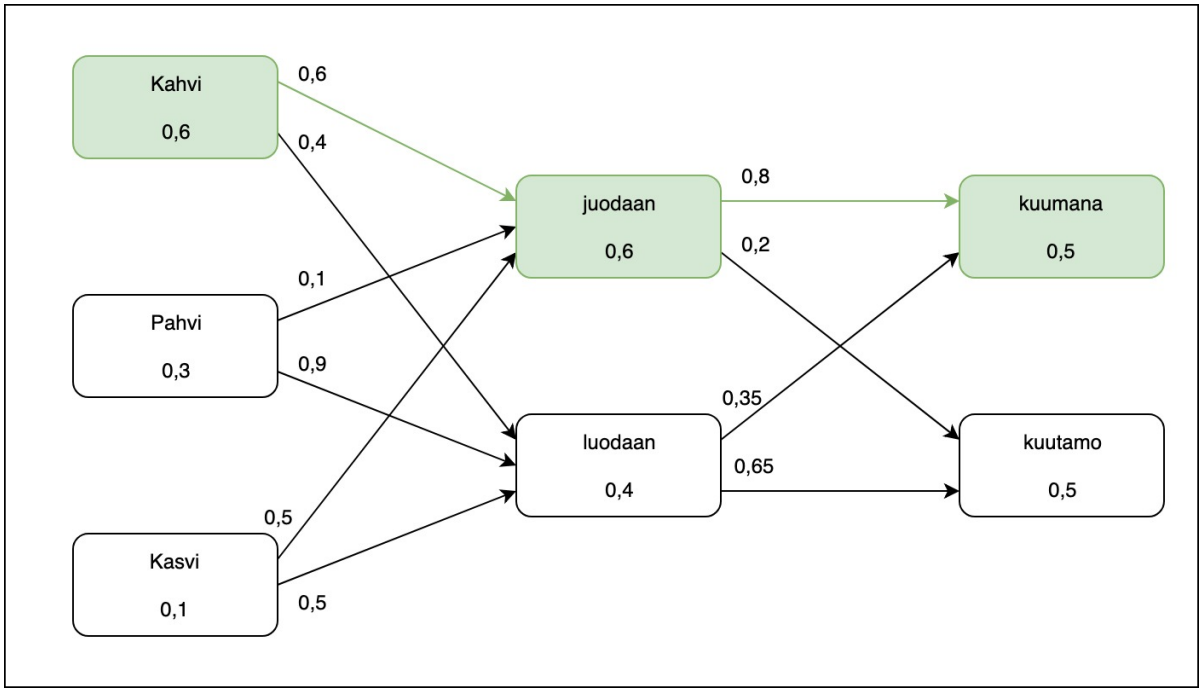
tinen malli on kätketty Markov-malli (HMM). HMM-malli otettiin yleisesti käyttöön puheentunnistuksessa 1980-luvulla, ja se on yleisessä käytössä vielä tänäkin päivänä. Äänteiden tilat näyttävät mallissa erilaisilta riippuen, missä vaiheessa äännettä tarkastellaan, ja siitä, mitkä ovat sen viereiset äänteet. Esimerkiksi /p/-äänne kuulostaa erilaiselta alussa kuin lopussa. Tämän takia HMM-malli koostuu puheentunnistuksessa yleensä kolmesta peräkkäisestä tilasta. (10.)

Foneemimallista saatuja äännesarjoja verrataan sanastomallin eli leksikon äännesarjoihin. Leksikkoon on kerätty paljon sanoja ja mistä äänteistä ne ovat muodostuneet. Leksikosta pyritään löytämään todennäköisin sana foneemimallista saatujen äännesarjojen perusteella. Leksikko on määritelty osittain manuaalisesti, sillä joitakin sanoja on vaikea muuttaa tekstistä automaattisesti äänteiksi. (7.)

Foneemimalli yhdessä sanastomallin kanssa tuottaa listan todennäköisiä sanoja. Kielimallin tehtävä on arvioida sanojen esiintymisen todennäköisyydet ja kuinka todennäköisesti peräkkäiset sanat esiintyvät keskenään lauseessa. Kielimallissa käytetään yleensä n-grammimallia, missä lasketaan sanayhdistelmien todennäköisyyksiä. Yhden sanan esiintymisen todennäköisyys tässä mallissa on riippuvainen n – 1 edellisestä sanasta. Puheentunnistuksessa käytössä on yleensä trigrammi (n = 3), joka ottaa huomioon sanan 2 edellistä sanaa. Lauseen kokonaistodennäköisyys muodostuu sanojen n-gram-todennäköisyyksien yhdistelmästä. Kielimallista on hyötyä esimerkiksi tilanteissa, joissa kahden akustisesti samankaltaisten sanojen todennäköisyydet ovat lähellä toisiaan. Esimerkiksi, jos järjestelmä on laskenut tietyn sanan olevan todennäköisesti joko ”kala” tai ”pala”, voidaan ottaa huomioon edeltävien sanojen semanttinen vaikutus. ”Järvessä ui kala” on todennäköisempi lause kuin ”Järvessä ui pala”. Malli oppii tekstissä esiintyvien sanojen esiintymistodennäköisyydet toisiinsa laajan opetusdatan avulla. (7.)

Dekooderin tarkoitus on löytää todennäköisin sanasarja järjestelmän todennäköisyysmalleista saatujen tuloksien perusteella. Todennäköisimmän sanasarjan löytämiseen käytetään erilaisia etsintäalgoritmeja, joista yksi käytetyin on Viterbi-algoritmi (10, s. 398). Kuvassa 3 on yksinkertaistettu esimerkki dekoodaus vaiheesta. Esimerkissä dekooderi on saanut tulokseksi lauseen: ”Kahvi juodaan kuumana”. Lauseen polku muodostuu sanojen todennäköisyyksistä ja peräkkäisten sanojen välisistä todennäköisyyk-

sistä. Ensimmäinen sana on ”Kahvi” 0,6 todennäköisyydellä ja todennäköisyys seuraavalle sanalle ”juodaan” on myös 0,6. Suurin polkujen todennäköisyyksien summa valitaan tulokseksi. Laskennan nopeuttamiseksi voidaan esimerkiksi määrittää jokin rajaarvo sanojen todennäköisyyksille, jonka alle jääviä sanoja ei oteta huomioon todennäköisintä polkua etsittäessä. Tämä nopeuttaa etsintää, mutta voi samalla heikentää järjestelmän tarkkuutta. (10.)

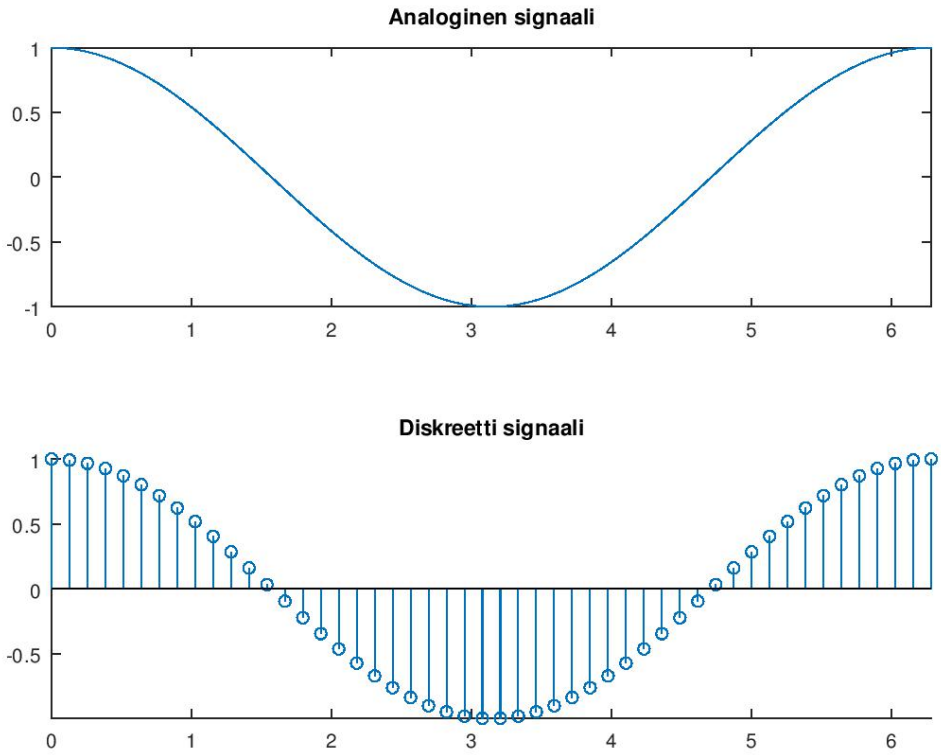


Kuva 3. Dekoodausvaiheessa pyritään löytämään polku, joka muodostuu suurimmasta todennäköisyyksien muodostamasta summasta.

2.3 Digitaalinen äänisignaalin käsittely

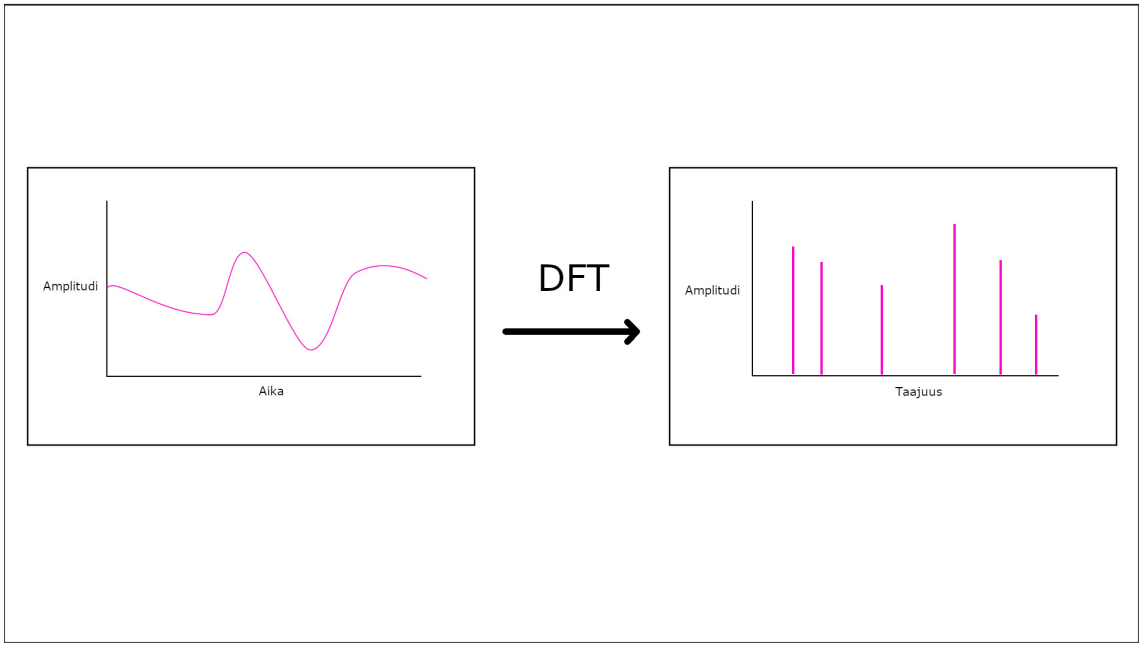
Jotta ääntä voitaisiin käsitellä digitaalisessa ympäristössä, on se ensin muutettava digitaaliseen muotoon analogisesta äänisignaalista. A/D-muunnoksessa (Analog/Digital) analoginen eli jatkuva-aikainen signaali muutetaan diskreettiaikaiseksi digitaaliseksi signaaliksi. Kuvassa 4 voidaan nähdä analogisen ja diskreetin aikasignaalin välinen ero. Analogisesta signaalista otetaan näytteitä tarpeeksi suurella näytteenottotaajuudella, ettei signaali vääristy muunnosvaiheessa. Nyquistin teoreeman mukaan näytteenottotaajuuden on oltava vähintään kaksi kertaa niin suuri kuin signaalissa esiintyvä korkein

taajuuskomponentti. Tällöin voidaan uudelleen muodostaa alkuperäistä signaalia vastaava signaali. (11, s. 1–3.)



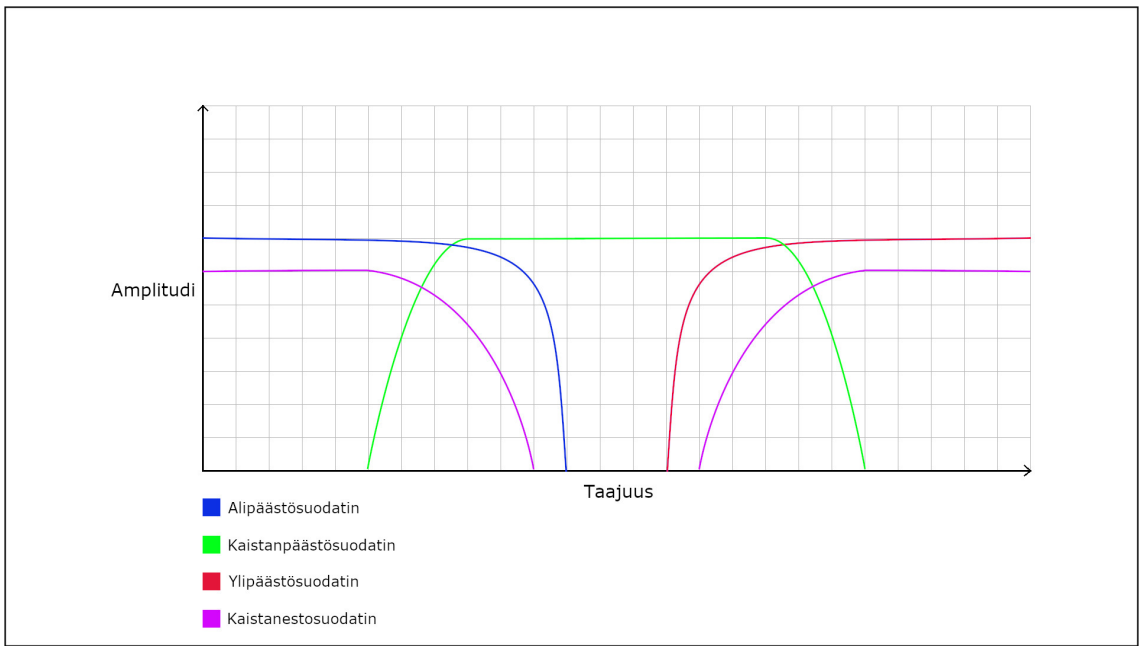
Kuva 4. Signaalin informaatiota katoaa vähän, kun näytteenottotaajuus on tarpeeksi korkea.

Signaalin taajuuksien analysointiin käytetään Fourier-muunnosta (kuva 5). Fourier-muunnoksen avulla voidaan signaalista selvittää, kuinka paljon kutakin taajuutta esiintyy signaalissa. Digitaaliselle signaalille käytetään yleensä DFT-muunnosta ja sen nopeaa FFT (Fast Fourier Transform) -versiota. Fourier-analyysissä signaali voidaan esittää eri taajuuksilla aaltoilevien kompleksisten eksponenttifunktioiden integraalina. Fourier-muunnoksen avulla voidaan selvittää kunkin aallon amplitudi ja vaihe. Monimutkaisten signaalien analysointi ja muokkaaminen on helpompaa, kun se on esitetty taajuustasossa. (11, s. 31–32; 12, s. 65–69.)



Kuva 5. Diskreetin Fourier-muunnoksen avulla voidaan laskea signaalissa esiintyvien taajuuksien vahvuudet.

Digitaalisten suodattimien käyttö on yleistä signaalin käsittelyssä. Digitaalisella suodattimella voidaan muokata signaalin taajuusvastetta niin, että signaalista suodatetaan pois turhat ja haitalliset taajuudet. Suodattimella voidaan myös vahvistaa haluttuja taajuusalueita. Äänisignaalista voidaan esimerkiksi suodattaa pois haitallinen taustakohina. Yleisiä suodattimia ovat muun muassa yli -ja alipäästösuodatin ja kaistanpäästösuodatin (kuva 6). Kaistanpäästösuodattimen avulla voidaan signaali rajoittaa tietylle kaista-alueelle suodattamalla pois matalat ja korkeat taajuudet. (13; 14.)

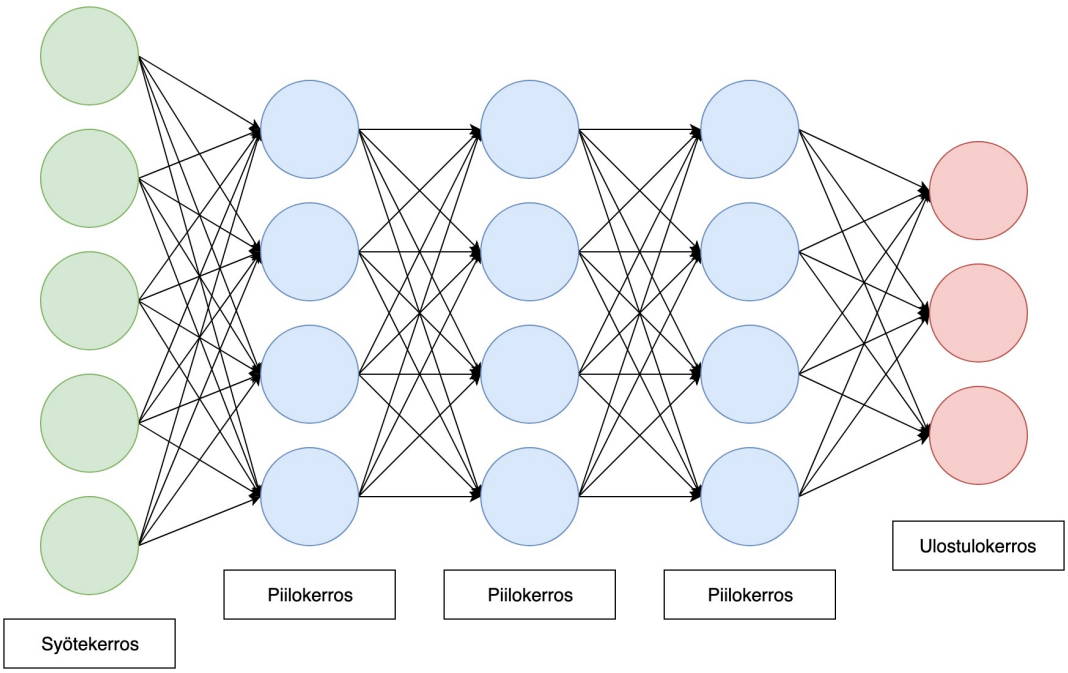


Kuva 6. Yleisimmät digitaaliset suodatin tyypit.

2.4 Syväoppiminen foneemimalleissa

Tietokoneiden laskentatehokkuuden ja saatavilla olevan puhedatan määrän kasvun myötä tekoälyn käyttö on yleistynyt puheentunnistamisjärjestelmissä. Syvien neuroverkkojen (DNN, Deep Neural Network) hyödyntäminen foneemimalleissa on parantanut puheentunnistamisjärjestelmien tarkkuutta merkittävästi. Niin sanotut DNN-HMM-mallit ovat syrjäyttäneet ennen yleisessä käytössä olleet Gaussin sekoitemalleihin perustuvat mallit (GMM-HMM). Syviin neuroverkkoihin perustuneiden mallien käyttöönoton vaikutusta puheentunnistamisessa voidaan verrata HMM-mallien käyttöönoton kaltaiseen teknologiaharppaukseen. (15, s. 299–309.)

Syväoppiminen on koneoppimisen osa-alue, jonka avulla voidaan mallintaa monimutkaisia tiedossa esiintyviä relaatioita. Syviä neuroverkkoja käytetään tunnistamaan piirteitä annetusta datasta. Syviä neuroverkkoja on käytetty apuna esimerkiksi kuvientunnistamisen ja itseohjautuvien autojen kaltaisissa sovelluksissa. Syvä neuroverkko (kuva 7) koostuu monikerroksisesta neuroverkostosta.



Kuva 7. Yksinkertaistettu esimerkki syvästä neuroverkosta.

Jokaisella neuronilla on oma lukuarvonsa, jonka avulla suoritetaan yksinkertainen usein epälineaarinen laskutoimitus. Neuroverkko voidaan kouluttaa ohjatusti opetusdatan avulla, joka esimerkiksi puheentunnistussovelluksessa on puheääni ja sitä vastaava teksti. Opetusdata syötetään malliin ja saatujen tuloksien perusteella säädetään neuroverkon neuronien lukuarvoja. Syvät neuroverkot voivat useissa tapauksissa olla hyvinkin isokokoisia, jonka seurauksena niiden kouluttaminen voi olla hidasta. Tietokoneiden laskentatehokkuuden paranemisen myötä syvien neuroverkkojen koulutusajat ovat lyhentyneet. (16.)

Perinteisen puheentunnistusjärjestelmän ohelle on kehittynyt niin sanottu End-to-End-puheentunnistusmenetelmä. End-to-End-menetelmän tarkoitus on korvata perinteisen puheentunnistusjärjestelmän eri komponentit yhdellä suurella neuroverkolla. Neuroverkko opetetaan muuttamaan sille syötetty puheääni tekstiksi ilman muita välivaiheita. End-to-End-neuroverkoilla on saavutettu hyviä puheentunnistustarkkuuksia, mutta sen ongelmana on ollut kyvyttömyys reaaliaikaiseen puheentunnistukseen, vaan mallille on täytynyt syöttää kokonainen äänitiedosto. RNN-T-malleilla (Recurrent neural network transducers) voidaan ohittaa tämä ongelma, koska mallilla voidaan tuottaa jatkuvaa ulostuloa sitä mukaan, kun sille syötetään puheääntä. (17.)

2.5 Suorituskyvyn mittaaminen

Puheentunnistusjärjestelmän suorituskykyä mitataan sen tarkkuuden ja nopeuden mukaan. Reaaliaikaista puheenkääntämistä tekstiksi vaaditaan esimerkiksi puhekäyttöliittymissä, joiden miellyttävän käytettävyyden yhtenä ehtona on käyttöliittymän nopeus. Esimerkiksi lääkärien sanelun muuttaminen tekstitiedostoksi ei välttämättä vaadi reaaliaikaisuutta, ja tällöin voidaan poistaa nopeussektori järjestelmän suorituskyvyn mittaamisesta. Nopeaa puheentunnistusta vaaditaan erilaisissa keskustelusovelluksissa, kuten puheella toimivissa chatboteissa. Ihmisten välisissä keskusteluissa luonnollinen puheenvuorojen väli on noin 0,6 sekuntia keskustelun luonteesta riippuen. Japanilaisessa tutkimuksessa (18) tutkittiin optimaalista vastausnopeutta keskustellessa käyttäjän kanssa. Tutkimuksen mukaan käyttäjät suosivat yhden sekunnin taukoa välittömän vastauksen sijaan. Robotin vastausnopeudessa voidaan soveltaa niin sanottua ”kahden sekunnin sääntöä”, jonka mukaan käyttöliittymän tulee vastata vähintään kahdessa sekunnissa käyttäjän antamaan syötteeseen miellyttävän käyttäjäkokemuksen saavuttamiseksi. Robotin vastauksen muodostamiseksi muut luonnollisen kielen prosessointiin liittyvät asiat saattavat myös viedä jonkin verran aikaa, joten puheentunnistus vaiheen tulee olla nopeaa. (18.)

Puheentunnistusjärjestelmän tarkkuutta mitataan yleensä sanojen virhemäärällä (WER, Word Error Rate). WER (kaava 4) vertaa puheentunnistusjärjestelmän tuottamaa tekstiä tekstiin, joka vastaa täydellisesti järjestelmälle annettua puhetta. Verrattavan tekstin sisältämien virheiden määrää verrataan tekstin sanojen kokonaismäärään, josta saadun tuloksen perusteella voidaan arvioida järjestelmän puheentunnistustarkkuus. (10.)

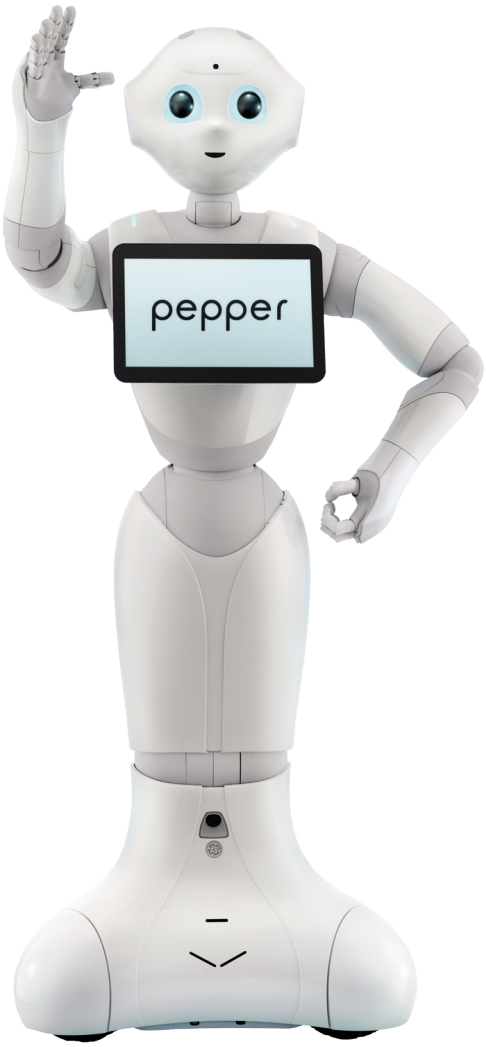
$$WER = 100 \times \frac{I+S+D}{N} \tag{4}$$

I on lisättyjen sanojen kokonaismäärä
S on vaihtuneitten sanojen kokonaismäärä
D on poistuneiden sanojen kokonaismäärä
N on alkuperäisten sanojen kokonaismäärä.

3 Puheentunnistus Pepper-robotissa

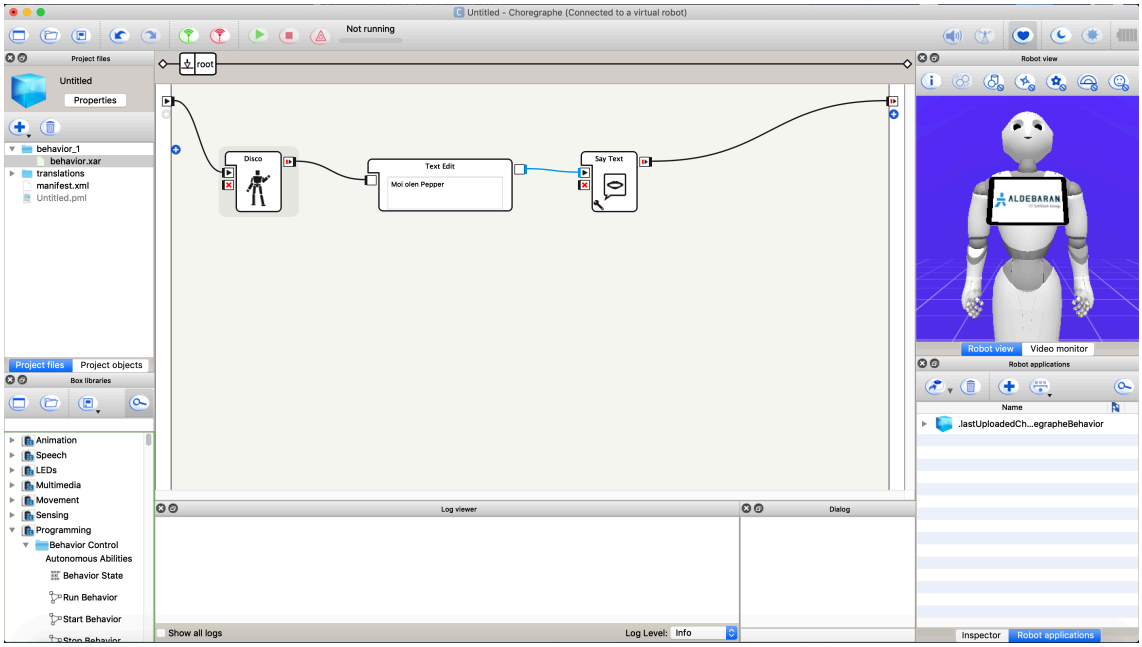
Pepper-robotti (kuva 8) on Softbank Robotics -yhtiön vuonna 2014 julkaisema humanoidirobotti. Pepper on seisoessaan noin 120 cm pitkä. Puhekäyttöliittymän lisäksi rintaan on liitetty kosketusnäyttö, jota voidaan myös käyttää vuorovaikutuksessa robotin kanssa. Softbank Robotics markkinoi Pepperin olevan maailman ensimmäinen humanoidirobotti, joka kykenee tunnistamaan ihmisten tunteita ja kasvoja. (19; 20.)

Tässä kappaleessa kerrotaan Pepper-robotin sisäänrakennetusta puheentunnistuksesta ja pohditaan sen mahdollisista heikkouksista.



Kuva 8. Pepper-humanoidirobotti (21)

Robotille voidaan kehittää toimintoja Softbank Roboticsin tarjoaman Choregraphe-kehitysympäristön (kuva 9) avulla. Choregraphen avulla voidaan robotille tehdä esimerkiksi animoituja liikkeitä, keskustelu dialogeja ja kustomoituja toiminnollisuuksia. Ohjelmointi kehitysympäristössä tapahtuu graafisesti yhdistelemällä pienempiä toimintoja sisältävien laatikoiden sisään- ja ulostuloja. Laatikoita voi myös muokata ja tehdä itse Python-ohjelmointikielen avulla. Robottia voidaan hallinnoida NAOqi Python-rajapinnan avulla, josta löytyvät funktiot yleisimmille robotin kehityksessä tarvittaville prosesseille.



Kuva 9. Esimerkki ohjelmoinnista Choregraphe-kehitysympäristössä.

Kuvassa 9 on yksinkertainen esimerkkiohjelma, joka kuvaa Choregraphessa tapahtuvaa graafista ohjelmointiprosessia. Ohjelman kulku alkaa vasemmasta reunasta ja jatkaa viivoja pitkin laatikosta laatikkoon aina ohjelman lopetukseen asti. Kuvan 9 ohjelmassa ohjelma kulkee ensimmäisenä ”Disco”-nimiseen laatikkoon, joka suorittaa animoidun tanssiliikkeen. Tämän jälkeen ohjelma siirtyy tekstilaatikkoon, jossa lukee: ”Moi olen Pepper”. Teksti välitetään laatikolle, jonka tehtävä on suorittaa sille syötetyn tekstin sanominen. Ohjelman suorittaminen lopetetaan, kun robotti on saanut sanottua tekstin loppuun.

3.1 Sisäänrakennettu puheentunnistus

Pepper käyttää puheentunnistuksessa NUANCE:n tarjoamaa puheentunnistusteknologiaa. Puheentunnistuksen perusominaisuudet ovat tuettu suomen kielessä (22). Pepperin sisäinen puheentunnistus tunnistaa vain sille etukäteen määriteltyjä sanoja ja lauserakenteita. (23.)

Ennen puheentunnistuksen käynnistämistä on järjestelmälle määriteltävä tunnistettava sanasto ja lauserakenteet. Korpuksen luomiseen voidaan käyttää esimerkiksi Dialog-tiedostoa, johon voidaan sanaston määrittämisen lisäksi määritellä robotin keskustelukäyttäytyminen. Dialog-tiedostossa määritellään robotin vastareaktio tiettyyn sanaan tai lauseeseen. Tiedosto käyttää QiChat-syntaksia (kuva 10) lausekkeiden ja vastareaktioiden määrittelyyn. Vastareaktio voi esimerkiksi olla sanallinen vastaus tai jonkin aliohjelman käynnistäminen. (23; 24.)

```
1  topic: ~EsimerkkiDialog()
2  language: fif
3
4  # Esimerkki konseptin määrityksestä.
5  concept:(tervehdys) [Moi, "Hei Pepper"]
6
7  # Esimerkki sanallisesta vastareaktiosta.
8  u: (~tervehdys) Moi, hauska nähdä sinua.
9
10 # Esimerkki "TellWeather" nimisen aliohjelman käynnistämisestä.
11 u: (Millainen sää on ulkona) $startProgram=TellWeather
12
```

Kuva 10. Esimerkki QiChat-syntaksista.

Jos kuvan 10 kaltainen Dialog-tiedosto määriteltäisiin puheentunnistusmoottorin käyttöön, kykenisi se tunnistamaan 3 lauseketta: ”moi”, ”mitä kuuluu” ja ”Millainen sää on ulkona”. Kun puhetta havaitaan, puheentunnistusjärjestelmä arvioi, millä todennäköisyydellä äänisignaali vastaa ennalta määriteltyjä lausekkeita. Kuvan 10 esimerkissä Pepper reagoisi käyttäjän tervehdykseen vastaamalla: ”Moi, hauska nähdä sinua.” (23.)

Dialog-tiedostoon voidaan tehdä keskustelurakenteita. Kuvassa 11 on esimerkki Dialog-tiedostoon tehdystä keskustelurakenteesta. Sulkujen sisään laitetaan käyttäjältä tunnistettava puhe ja sulkujen jälkeen kirjoitetaan robotin vastaus puheeseen. Esimerkissä käyttäjä on kysynyt robotilta, pitääkö tämä eläimistä. Robotin vastauksessa esitetään vastakysymys, joka kysyy käyttäjältä, onko hänellä koiraa tai kissaa. Tämän jälkeen robotti odottaa käyttäjältä jotain alisääntöjen (u1) kaltaista määriteltyä vastausta. Keskustelu etenee alisäännöistä toiseen käyttäjän vastauksien mukaan.

```
topic: ~Esimerkki_keskustelurakenne()
language: fif

u:(Pidätkö eläimistä) Kyllä pidän. Onko sinulla koiraa tai kissaa?
    u1:(koira) Onko koirasi iso vai pieni?
        u2:(iso) Hui! pelkään isoja koiria.
        u2:(pieni) Pienet koirat ovat söpöjä.
    u1:(kissa) Onko kissasi hyvä pyydystämään hiiriä?
        u2:(kyllä) Vau! Siitä taidosta on varmasti hyötyä kissalle.
        u2:(ei) Voisin opettaa kissaasi pyydystämään hiiriä.
    u1:(ei kumpaakaan) Ei minullakaan, koska olen allerginen molemmille. Heh!

    |
proposal: Seuraatko paljon urheilua?
    u1:(kyllä) Kiva! niin minäkin.
    u1:(ei) Selvä homma!
```

Kuva 11. Esimerkki Dialog-tiedoston keskustelurakenteesta.

QiChat-syntaksiin kuuluvan ”proposal”-termin avulla voidaan esimerkiksi tehdä aloite keskustelussa käyttäjän kanssa. Aloitteen tekemisen avulla saadaan tehtyä robotista inhimillisempi keskustelutilanteissa. Kun robotti tekee aloitteen keskustelussa, jää se odottamaan alisääntöihin määriteltyjä vastausvaihtoehtoja.

Vaikka keskustelurakenteiden tekeminen Dialog-tiedostoihin on helppoa, voi se laajassa mittakaavassa olla kuitenkin työlästä. Lauserakenteiden määrittely QiChat-syntaksin avulla helpottuu, mutta sen avulla on silti mahdotonta listata kaikkia mahdollisia lauserakenteita Dialog-tiedostoon.

3.2 Puheentunnistuksen heikkoudet

Hyviä puolia robotin sisäisessä puheentunnistuksessa on sen nopeus ja melko hyvä tarkkuus ennalta määritellyn sanaston tunnistuksessa. Puheentunnistuksen nopeus ei kärsi merkittävästi sanaston kasvaessa. Sanaston määrittelyssä pitää välttää liian samankaltaisten lauseiden määrittelyä. Robotin voi olla vaikea tunnistaa kahta samankaltaista lausetta toisistaan. Robotti tunnistaa puheen hyvin, kun puhuja seisoo suoraan robotin edessä, noin metrin päässä siitä. Sisäinen puheentunnistus toimii huomattavasti paremmin hiljaisessa kuin meluisassa ympäristössä.

Suurin puute robotin omassa puheentunnistuksessa on sen kykenemättömyys vapaaseen puheentunnistamiseen. Tämä tarkoittaa sitä, että robotti tunnistaa vain sanoja ja lauseita, jotka on ennalta määritelty sen sanastoon. Vapaa puheentunnistus on hyödyllistä esimerkiksi puheen merkityksen tunnistamisessa keskusteluissa, koska ei voida etukäteen tietää, mitä ja miten keskustelussa oleva henkilö puhuu. Myös erisnimien erottelussa hyödytään vapaasta puheentunnistuksesta, sillä tällöin jokaista erisnimeä ei tarvitse erikseen etukäteen määritellä sanastoon. Esimerkiksi kun halutaan hakea puheella tietyn kaupungin säätietoja, kaupunki-parametrin saamiseksi joudutaan robotin sisäisessä puheentunnistuksessa määrittelemään sanastoon kaikki mahdolliset kaupungit. Vapaan puheentunnistuksen puute tekee keskustelun robotin kanssa rajallisemmaksi, koska tällöin useissa tapauksissa robotille mahdollisten sanottavien sanojen määrä on pieni.

Pepper on tarkoitettu verrattain hiljaisiin sisätiloihin, kuten yritysten hiljaisiin aulatiloihin palvelemaan asiakkaita. Pepperillä on vaikeuksia tunnistaa puhetta taustahälinästä tapahtumissa, missä on paljon ihmisiä. Robotti keskittyy kuuntelemaan katseellaan seuraamaansa henkilöä, ja väenpaljoudessa se saattaa kadottaa henkilön katsepiiristään, jolloin puheentunnistaminen saattaa huonontua. Ihminen pystyy tunnistaan todella hyvin puhetta meluisissakin ympäristöissä, mikä voi koneelle tuottaa paljon ongelmia (6, s. 340). Verrattuna ihmiseen koneen on vaikea tunnistaa taustapuhetta tunnistettavasta puheesta, koska ne muistuttavat äänellisesti toisiaan.

4 Puheentunnistuksen parantaminen

4.1 Ulkoinen puheentunnistus

Pepperin sisäisen puheentunnistuksen heikkouksien takia tutkittiin kolmansien osapuolten tarjoamia ratkaisuja. Palvelun kriteereinä olivat suomen kielen tuki ja edullisuus. Palvelun tulisi myös kyetä vapaaseen puheentunnistukseen, jotta se olisi järkevä ottaa käyttöön sisäisen puheentunnistuksen sijaan. Suomen kieli on markkina-alueena globaalisti verrattuna pieni. Investointi vähän puhuttuihin kieliin ei välttämättä ole taloudellisesti kannattavaa, ja siksi monet puheentunnistuspalvelut ovat usein tarjolla vain paljon puhutuilla kielillä. Puheentunnistuksen saatavuus suomen kielellä rajasi mahdollisia vaihtoehtoja.

Vaihtoehtona robotin sisäiselle puheentunnistukselle lähdettiin testaamaan Google Cloudin tarjoamia puheentunnistuspalveluita. Googlen Speech-To-Text (STT) -puheentunnistuspalvelu valittiin, koska siinä oli suomen kielen tuki ja lyhyellä testauksella puheentunnistustarkkuus vaikutti tyydyttävältä. Palvelu tarjoaa 60 minuuttia ilmaista puheenkäännöstä kuukausittain. Google Cloud Platform tarjoaa myös 300 dollaria palvelun käyttöön 12 kuukautta kestävällä kokeilujaksolla. Palvelun käytön testauksesta ei siis aiheutunut ylimääräisiä kuluja. Googlen puheentunnistuksen hyviin puoliin kuuluu sen kyvykkyys vapaaseen puheentunnistukseen. Näin ollen tunnistettavien sanojen määrä on laaja, ja palvelu kykenee tunnistamaan myös vieraskielisiä sanoja. Vapaa puheentunnistus on hyödyllistä robotissa esimerkiksi siksi, koska se laajentaa sanastoa, mitä robotille voidaan sanoa.

Ensimmäisenä kokeiltiin puheentunnistusta kokonaisesta äänitiedostosta. Google STT -rajapinnalle lähetetään kutsu, joka sisältää robotilla nauhoitetun äänitiedoston. Vastauksena rajapintakutsuun palvelulta saadaan oletettu puheen tekstiversio. Vastaus sisältää myös käännöksen luotettavuuden todennäköisyyden, joka voidaan ottaa huomioon esimerkiksi keskusteluissa. Jos todennäköisyys tekstin käännöksen luotettavuudelle on liian pieni, voi robotti esimerkiksi vastata, ettei ymmärtänyt puhujan sanomaa lausetta. Palvelusta saadulle vastaukselle voidaan tehdä luonnollisen kielen käsittelyä, jonka avulla voidaan vastaustekstistä löytää puhutun lauseen tarkoitus ja siinä mahdollisesti esiintyvät entiteetit. Lauseen tarkoituksen ja siinä olevien entiteettien avulla voidaan määrittää robotin reagointi puheeseen. Rajapintaa oli helppo käyttää ja palvelusta saatiin

kohtalaisen tarkkoja vastauksia. Palvelun heikkoutena voidaan pitää sen käytöstä aiheutuvaa verkkoviivettä. Vastauksen saaminen palvelusta voi kestää useita sekunteja. Keskustelun sujuvuus kärsii, kun vastakkainen osapuoli joutuu odottamaan liian kauan vastausta hänen puheeseensa.

Verkkoviiveen minimoimiseksi seuraavaksi testattiin Googlen puheentunnistuspalvelun versiota, jossa lähetetään äänivirtaa palvelulle. Palvelulle välitetään ääntä esimerkiksi suoraan mikrofonista reaaliajassa. Palvelu lähettää takaisin vastauksia sitä mukaan, kun ääntä on saatu käsiteltyä taustapalvelussa. Vastauksessa välitetään myös tieto siitä, kun palvelu on havainnut mahdollisen lauseen päättymisen. Tämä tieto on hyödyllistä esimerkiksi, kun halutaan lopettaa äänisignaalin lähettäminen palveluun. Äänen lähettäminen palvelulle on tällä hetkellä mahdollista vain gPRC:n välityksellä (25). Palvelun käyttöönotto oli helppoa, ja vastauksia saatiin takaisin palvelulta huomattavasti nopeammin kuin lähettämällä kokonainen äänitiedosto Google STT -palveluun. Äänivirtaa lähetettäessä palvelulta saadut vastaukset eivät olleet yhtä tarkkoja kuin kokonaisia tiedostoja lähetettäessä. Ajoittain palvelu jumittui ja vastauksien saaminen palvelulta keskeytyi. Tällöin jouduttiin muodostamaan uusi yhteys palveluun. Äänivirtaa voidaan yhtäjaksoisesti lähettää palvelulle enintään minuutin ajan. Keskustelusovelluksissa tämä ei ole usein haittatekijä, koska puhujan puheenvuoro vuoropuhelussa kestää harvoin yhtäjaksoisesti yli minuuttia. Äänivirran lähettäminen palveluun on parempi vaihtoehto ulkoiseksi puheentunnistukseksi keskustelusovelluksissa sen reaaliaikaisuuden myötä. Tämän takia lähdettiin pääasiallisesti testaamaan äänivirran lähettämistä palvelulle Pepper-robotin vaihtoehtoiseksi puheentunnistukseksi.

Äänivirtaa palveluun lähettävä puheentunnistusversio liitettiin Pepperin muuhun ohjelmistoon taustalla käynnissä olevana aliohjelmana. Ohjelma tallentaa robotin mikrofonista saatua ääntä puskuriin. Kun äänenvoimakkuus nousee yli tietyn kynnysarvon, aloitetaan äänen lähettäminen puskurista Googlen taustapalveluun. Puskurista lähetetään äänidataa, joka alkaa hieman ennen hetkeä, kun äänenvoimakkuuden kynnysarvo ylitettiin. Näin voidaan estää ensimmäisen sanan jääminen pois käännöksestä, koska yhteyden muodostaminen Googlen taustapalvelun kanssa saattaa aiheuttaa pientä viivettä ohjelmassa. Ääntä lähetetään taustapalveluun niin kauan, kunnes palvelun vastauksessa ilmenee, että puhuttu lause on päättynyt. Tämän jälkeen vastauksista valitaan todennäköisin puheen käännös, joka julkaistaan käytettäväksi muille robotin ohjelmiston

osille. Ohjelmiston muut osat voivat käyttää puheenkäännöstä esimerkiksi erilaisten toimintojen käynnistämiseen.

Pepperillä on yhteensä 4 mikrofonia, jotka sijaitsevat sen päässä. Mikrofonien äänityksen taajuusalue on 100–10 000 Hz. Robotin päässä on myös tuulettimia, jotka tuottavat äänisignaaliin puheentunnistukselle haitallista taustamelua. Tuulettimien tuottama taustamelu on todennäköisesti otettu huomioon robotin sisäisessä puheentunnistusjärjestelmässä, mutta ulkoisia palvelua käytettäessä taustamelusta voi aiheutua huomattavaa haittaa. Ulkoista puheentunnistusta lähdettiin parantamaan suodattamalla tuulettimien tuottamaa taustamelua pois äänisignaalista. Tarkoitus oli testata, vaikuttaako taustamelun vähentäminen äänisignaalista myönteisesti Googlelta saatuihin puheentunnistustuloksiin. Tuloksien tarkkuutta mitattiin WER-sanavirhetestillä.

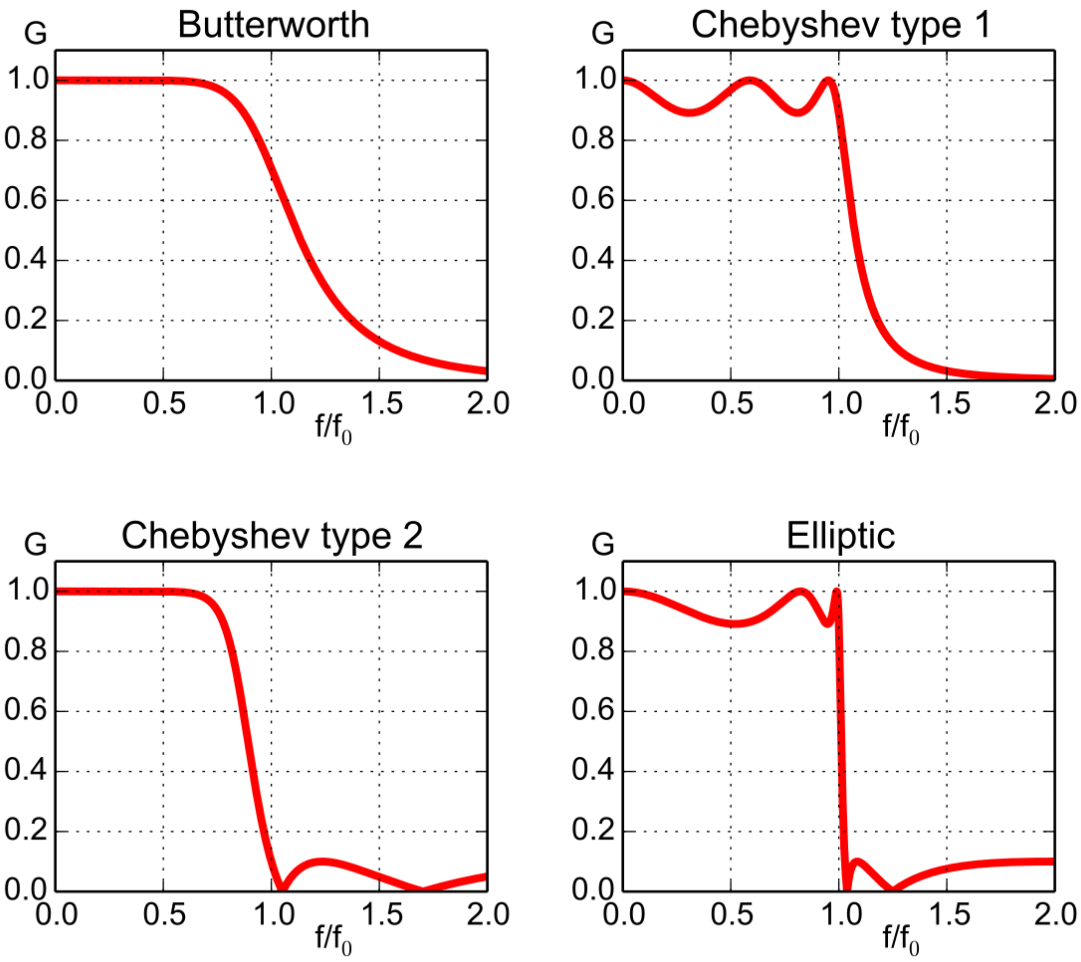
4.2 Taustamelun vähentäminen

Äänisignaalin käsittelyn tarkoituksena oli vähentää robotin tuulettimien tuottamaa melua signaalissa, ja tämän seurauksena mahdollisesti parantaa Googlen puheentunnistuksen tarkkuutta.

Ihmisen puhe sijaitsee puhujasta riippuen noin 100–17 000 Hz:n välisellä taajuusalueella (26). On havaittu, että suurin osa puheentunnistuksen kannalta tärkeästä informaatiosta sijaitsee noin 300–3 400 Hz:n välisellä taajuusalueella (27). Kaistanpäästösuodattimella voidaan rajata äänisignaali tietylle taajuusalueelle. Idea kaistanpäästösuodattimen käytöstä taustamelun vähentämisessä tuli siitä, että suurin osa puheen tärkeästä informaatiosta löytyy suhteellisen kapealta taajuusalueelta. Kokeilun tarkoituksena oli selvittää, vaikuttaako kaistanpäästösuodattimen käyttö myönteisesti Googlen puheentunnistuksen tarkkuuteen.

Toteutuksessa kokeiltiin kaistanpäästösuodattimia eri raja-arvoilla. Suodatintyyppinä käytettiin toisen asteen Butterworth-suodatinta (28). Butterworth-suodatintyyppiä käytetään paljon äänenkäsittelyssä, koska suodattimen taajuusvaste on melko suora verrattuna joihinkin muihin yleisesti käytettyihin suodatintyyppeihin (kuva 12). Kaistanpäästösuodattimen läpimenevänä taajuuskaistana kokeiltiin 300–2 500 Hz:n ja 200–3 400

Hz:n välisiä alueita. Taajuuskaistan ulkopuolelle jääneet taajuudet suodatetaan pois äänisignaalista. Eri taajuuskaistoja kokeiltiin sen takia, että voitaisiin mahdollisesti löytää parhaat ala- ja yläarvot suodattimelle puheentunnistuksen kannalta. (28.)



Kuva 12. Eri suodatintyyppien taajuusvasteita. (29)

Toisena tapana taustamelun vähentämiseen käytettiin häiriönpoistoalgoritmia (30), joka perustuu taustamelun poistoon Fourier-analyysin avulla. Algoritmiin syötetään kaksi signaalia: käsiteltävä äänisignaali ja poistettavaa häiriötä sisältävä äänisignaali. Fourier-analyysillä pyritään selvittämään häiriösignaalin taajuusspektri, jonka avulla voidaan hiljentää häiriötaajuuksia käsiteltävästä äänisignaalista. Algoritmi soveltuu staattisen häiriöäänen poistoon. Algoritmia varten nauhoitettiin lyhyt ääninäyte robotin tuulettimien tuottamasta häiriöäänestä.

Algoritmin ensimmäisessä vaiheessa pilkotun häiriösignaalin osille lasketaan FFT-muunnos. Tämän jälkeen FFT-muunnoksista lasketaan statistiikat, kuten keskimääräinen teho, jotka taulukoidaan jokaiselle eri taajuusalueelle. Taajuusalueiden oletusmäärä algoritmissa on 1 025 aluetta. Statistiikan ja algoritmille säädettyjen arvojen avulla lasketaan tehon raja-arvo jokaiselle taajuusalueelle. Sitten lasketaan muutettavan äänisignaalin FFT-muunnos, jonka jälkeen signaalin taajuusalueiden voimakkuuksia verrataan häiriösignaalista laskettuihin raja-arvoihin. Jos taajuusalueen voimakkuus jää alle raja-arvon, vaimennetaan taajuusaluetta signaalin sen hetkisessä osassa. Seuraavaksi voimakkuuksien muutoksia tasoitetaan, jotta muutokset taajuusalueiden voimakkuuksissa eivät olisi niin jyrkkiä. Viimeiseksi taajuusalueiden voimakkuuksien muutokset tehdään muutettavan signaalin FFT-muunnokselle, jonka jälkeen käänteisellä FFT-muunnoksella saadaan ulos käsitelty äänisignaali. (30; 31.)

4.3 Testausmenetelmä

Googlen puheentunnistuspalvelun tarkkuutta ja äänenkäsittelyn vaikutusta siihen päätettiin testata vastaustekstien WER-tuloksia vertailemalla. Ulkoisen puheentunnistuksen suorituskyvyn testauksella oli tarkoitus selvittää, onko ulkoista puheentunnistusta järkevä ottaa käyttöön robotissa. Haluttiin myös selvittää, onko äänisignaalin käsittelyllä myönteistä vaikutusta puheentunnistuksen tarkkuuteen. Oli myös mielenkiintoista nähdä, onko puheäänen lähetystavalla merkitystä Googlen puheentunnistuspalvelun suorituskykyyn.

Testiä varten nauhoitettiin noin 10 minuuttia miespuolisen henkilön puhetta. Puhe nauhoitettiin robotin mikrofonia käyttäen ja puhe-etäisyys mikrofonista oli noin metrin. Äänityksessä käytetty huone oli akustisilta ominaisuuksiltaan hiljainen ja kaikumaton. Puhe jaettiin noin 30 sekuntia pitkiin raakoihin äänitiedostoihin, jotta puheen äänivirran lähetystä palveluun voitaisiin testata. Alun perin puhe oli jaettu hieman alle minuutin kestäviin osiin, mutta palvelulta saadut tulokset äänivirtaa lähetettäessä olivat tällöin todella heikkoja. Lyhyemmät puheosuudet ovat myös lähempänä keskustelun kaltaista pituutta, koska ihmisen puheenvuoro dialogin aikana kestää harvoin lähemmäs minuuttia.

Googlen puheentunnistuksesta saadut vastaukset tallennettiin tekstitiedostoihin, joita verrattiin vastaaviin alkuperäisiin puhuttuihin teksteihin. Testissä käytettiin Jiwer Python-kirjastoa WER-tuloksien laskemiseen.

WER-tulokset laskettiin käsittelemättömistä ja käsitellyistä puheista. Tulokset laskettiin erikseen myös kokonaisten äänitiedostojen sekä äänivirran lähettämisestä palveluun, jotta niiden suorituskykyä voitaisiin vertailla keskenään.

5 Tulokset

Työssä tehdyn WER-testin tuloksista voidaan jonkin verran päätellä Googlen puheentunnistuksen toimivuudesta robotin kanssa. Google on itse ilmoittanut vuonna 2017 englanninkielisen puheentunnistuksen yltäneen 4,7 %:n WER-tulokseen (32). Noin alhaiseen tulokseen on todennäköisesti päästy käyttämällä todella rajattua opetus- ja testausaineistoa järjestelmän opetuksessa ja testauksessa. Voidaan myös olettaa, että testissä käytetty puhe on nauhoitettu meluttomassa olosuhteissa, laadukkaalla mikrofonilla. Tämän työn testissä saadut tulokset ovat huomattavasti heikompia kuin Googlen testin ilmoitettu 4,7 %:n WER-tulos. Testin WER-tuloksia ovat heikentäneet esimerkiksi robotin mikrofonin laatu, äänisignaalissa esiintyvä taustamelu ja satunnaisesti valittu teksti puhetta varten. Testissä saatuihin tuloksiin on voinut myös vaikuttaa puhujan puhetyyli sekä suomen kielen käyttö testissä. (32.)

Kun puheentunnistusta käytetään normaaleissa käyttöolosuhteissa, ovat tässä työssä saadut WER-tulokset (kuva 13) lähempänä todenmukaisuutta kuin 4,7 %:n tulos. Normaaleissa käyttöolosuhteissa taustamelu voi olla hyvinkin voimakasta ja mikrofonin laatu voi vaihdella käytössä olevan laitteen myötä. Testissä käytetty puhe nauhoitettiin hiljaisessa tilassa ja robotin tuulettimet olivat ainoa taustamelua tuottava lähde. Voidaan olettaa, että testissä saadut WER-tulokset meluisassa tilassa olisivat olleet vielä heikompia. Tässä työssä tehdyn testin tuloksien luotettavuutta voidaan myös kyseenalaistaa, koska testissä käytettyä puhetta oli suhteellisen vähän ja puhujana käytettiin vain yhtä henkilöä. Luotettavampia tuloksia saataisiin kasvattamalla testipuheen määrää ja lisäämällä testiin useamman henkilön puhetta.

Testissä saatujen tulosten perusteella kokonaisten äänitiedostojen lähettäminen Googlen puheentunnistusjärjestelmälle on tarkempaa kuin reaaliaikaisen puheen äänivirran lähettäminen. Käsittelemättömän äänitiedoston lähettäminen Googlen palveluun saavutti keskimäärin 28 %:n WER-tuloksen, kun taas vastaava tulos käsittelemätöntä äänivirtaa lähetettäessä oli 37 %:n, joka on 9 prosenttiyksikköä huonompi tulos.

Googlen puheentunnistusjärjestelmän WER-testin tulokset*

Puhetiedosto	Kokonaisen tiedoston lähettäminen Googlen puheentunnistukselle			Google STT streaming versio			
	Käsittelemätön	Kaistanpäästösuodatin 200-3400 kHz	FFT-suodatin	Käsittelemätön	Kaistanpäästösuodatin 200-3400 kHz	Kaistanpäästösuodatin 300-2500 kHz	FFT-suodatin
Keskiarvo**	**28**	**30**	**31**	**37**	**36**	**38**	**32**
1	31	27	33	31	29	24	27
2	26	26	22	22	30	44	22
3	36	33	38	33	33	33	36
4	31	39	45	31	24	45	39
5	32	30	38	30	30	30	35
6	16	16	19	19	19	16	19
7	23	26	26	47	40	40	33
8	18	21	21	18	21	33	28
9	32	30	32	62	60	68	35
10	36	40	31	33	38	36	33
11	33	30	30	37	42	28	49
12	25	25	29	68	75	68	29
13	29	50	35	32	38	32	38
14	9	9	9	40	40	40	9
15	29	39	29	29	29	32	29
16	21	26	32	26	26	26	21
17	31	33	38	28	33	59	38
18	27	31	33	65	33	33	35
19	31	31	33	48	40	38	40
20	23	40	30	40	40	40	30
21	33	33	33	42	38	44	31

* Tulokset on ilmoitettu prosentteina
** Kaikista puhetiedostoista saatujen WER-tulosten painotettu keskiarvo. Keskiarvon painotus ottaa huomioon puhetiedoston alkuperäisen sanamäärän.

Kuva 13. Googlen puheentunnistuksesta saadut WER-tulokset.

Kokonaisen äänitiedoston käsitteleminen ennen sen lähettämistä Googlen palveluun vaikutti negatiivisesti WER-tuloksiin. Oletettu syy tälle on se, että äänitiedoston käsittelyssä häviää jotain puheentunnistuksen kannalta tärkeää informaatiota. Kaistanpäästösuodattimen vaikutukset WER-tuloksiin olivat olemattomat sekä kokonaisen äänitiedoston puheentunnistuksessa, että äänivirtaa lähetettäessä. Yksi syy tälle on se, että tuulettimista syntyvässä äänessä on paljon taajuuksia, jotka osuvat puheen taajuusalueelle. Täten ne myös pääsevät läpi kaistanpäästösuodattimesta. Voi olla myös, että Googlen puheentunnistus käyttää järjestelmässään jonkin tyyppistä kaistanpäästösuodatinta, jolloin signaalin suodattaminen ei vaikuta tuloksiin.

Häiriönpoistoalgoritmin käytöllä äänivirtaa lähetettäessä saatiin varteenotettavia parannuksia keskimääräisessä WER-tuloksessa. Käyttämällä häiriönpoistoalgoritmia äänivirtaa lähetettäessä tulos parani käsittelemättömän signaalin 37 prosentin WER-tuloksesta 32 prosenttiin. Voidaan siis arvioida, että häiriönpoistoalgoritmin käytöstä on jonkin asteista hyötyä, kun lähetetään puheen äänivirtaa Googlen palveluun. Häiriönpoistoalgoritmin käytöllä lähestytään käsittelemättömän kokonaisen äänitiedoston puheentunnistuksen testituloksen tarkkuutta. Jo kuuntelemalla häiriönpoistoalgoritmin avulla käsiteltyä ääntä voidaan havaita, että menetelmällä on saatu vähennettyä tuulettimista johtuvaa häiriääntä. Kaikkea häiriääntä ei algoritmilla kuitenkaan saatu poistettua äänestä.

WER-tuloksista voidaan siis päätellä, että kokonaista äänitiedostoa ei kannata käsitellä testatuilla menetelmillä ennen sen lähettämistä Googlen puheentunnistuspalveluun. Havaittiin myös, että äänen suodattaminen testatuilla kaistanpäästösuodattimen taajuusalueilla ei vaikuttanut merkittävästi puheentunnistustarkkuuteen. Häiriönpoistoalgoritmin käyttämisellä äänivirtaa lähetettäessä puheentunnistukseen saatiin lupaavia tuloksia, joten siitä olisi mahdollisesti järkevä tehdä jatkotutkimuksia. Säätämällä algoritmiin annettavia parametreja voidaan mahdollisesti parantaa Googlen palvelusta saatuja tuloksia.

WER-testiin olisi ollut mielenkiintoista ottaa mukaan jollain muulla laitteella tallennettua puhetta. Esimerkiksi oltaisiin voitu tallentaa puhe samanaikaisesti puhelimella ja vertailla, kuinka paljon laitteissa olevat mikrofonit vaikuttavat puheentunnistustulokseen. Olisi ollut myös mielenkiintoista nähdä, onko miesten ja naisten puheäänten eroilla merkittävää vaikutusta puheentunnistuksen tarkkuuden. Tämän asian tutkimiseen olisi kuitenkin tarvittu suuri määrä eri koehenkilöiden puhetta, jotta tutkimuksesta saadut tulokset olisivat luetettavia ja todenmukaisia.

Yhtenä työn tarkoituksena oli tutkia ulkoisen puheentunnistuksen käyttöönoton kannattavuutta Pepper-robotissa. Jotta tätä voitaisiin arvioida, täytyy verrata ulkoisen puheentunnistusjärjestelmän nopeutta, tarkkuutta ja vapaasta puheentunnistuksesta saatuja hyötyjä Pepperin sisäiseen puheentunnistukseen. Nopeus on tärkeä ominaisuus puheentunnistusjärjestelmässä, jota käytetään keskustelusovelluksissa. Pepperin sisäinen puheentunnistus voittaa ulkoiset järjestelmät nopeudessaan. Vaikka äänivirran kääntä-

minen tekstiksi Googlen palvelussa on merkittävästi nopeampaa kuin kokonaista tiedostoa käännettäessä, ei senkään nopeus ole vielä sillä tasolla, että sitä voitaisiin käyttää pääasiallisena puheentunnistusjärjestelmänä robotissa.

Sisäisen puheentunnistuksen puheenkääntämisen tarkkuus on melko hyvä, mutta sen tarkkuutta arvioitaessa on huomioitava, että se tunnistaa vain sille ennalta rajattuja sanoja ja lauserakenteita. Sen takia sisäistä puheentunnistusta ei voitu ottaa mukaan WER-testiin. Googlen puheentunnistuksen hyötynä on sen kyvykkyys vapaaseen puheentunnistukseen. Sen avulla voidaan tunnistaa laajempi määrä sanoja ja lauserakenteita puheesta. Tämä on erittäin hyödyllistä keskustelusovelluksissa, koska ei voida etukäteen tietää, mitä puhuja aikoo sanoa.

Saatujen tuloksien perusteella ei mielestäni kannata kokonaan korvata Pepperin sisäistä puheentunnistusta Googlen ulkoisella puheentunnistuksella. Ulkoisen puheentunnistuksen käyttäminen kuitenkin yhdessä sisäisen puheentunnistuksen kanssa voisi olla hyödyllistä. Puheentunnistusjärjestelmien yhteiskäyttö voisi tapahtua niin, että pääasiallinen puheentunnistus tapahtuisi sisäistä järjestelmää käyttäen. Jos sisäisestä puheentunnistuksesta saadun tunnistuksen todennäköisyys ei ole tarpeeksi suuri, hyödynnettäisiin ulkoisesta puheentunnistuksesta saatua tulosta. Ulkoista puheentunnistusta voitaisiin käyttää myös tapauksissa, missä käyttäjän puheesta tarvitaan saada jokin kontekstimuuttuja. Konteksti muuttuja voi olla esimerkiksi jokin erisnimi, kuten kaupunki.

Kun katsotaan testissä saatuja tuloksia laajemmasta näkökulmasta, voidaan sanoa, että puheentunnistuksessa käytettävä tekniikka varsinkaan suomenkielellä ei vielä läheskään saavuttanut sen täydellistä potentiaaliaan. Tämä huomio voidaan tehdä, jos oletetaan, että Googlen tarjoama puheentunnistus on yksi alan parhaita. Parantamisen varaa jäi sekä tarkkuuden että nopeuden osa-alueilla. On kuitenkin todennäköistä, että puheentunnistus tulee parantumaan tekniikan kehittyessä ja puhekäyttöliittymien lisääntymisen vaikutuksesta tulevaisuudessa.

6 Yhteenveto

Työn tarkoituksena oli tutustua puheentunnistusjärjestelmien toimintaperiaatteisiin ja tutkia, voidaanko ulkoisella palvelulla parantaa Pepper-robotin puheentunnistusta. Pyrittiin löytämään suomenkielinen ratkaisu, joka kykenee vapaaseen puheentunnistukseen. Ulkoisen puheentunnistuspalvelun tulisi olla tarpeeksi nopea ja tarkka, että se olisi hyödyllistä ottaa käyttöön Pepper-robotin sisäisen puheentunnistuksen sijaan. Testattavaksi ulkoiseksi palveluksi valittiin Google Cloudin tarjoama puheentunnistuspalvelu.

Työssä selvisi, että kokonaisen äänitiedoston lähettäminen Googlen pilvipalveluun on liian hidasta, jotta robotissa voisi käyttää sitä tapaa puheentunnistuksessa. Robotin kanssa luontevan keskustelun saavuttamiseksi käyttäjä ei voi odottaa robotin vastausta liian kauan. Tämän takia Googlen puheentunnistuspalvelusta valittiin käyttöön sille äänivirtaa lähettävä versio, joka osoittautui nopeammaksi kuin kokonaisen äänitiedoston lähettäminen palveluun. Äänivirran lähettäminen palveluun valikoitui paremmaksi vaihtoehdoksi robotin ulkoiselle puheentunnistukselle sen nopeuden takia.

Googlen puheentunnistuksesta saatuja tuloksia testattiin WER-menetelmää käyttäen. Puheentunnistusta testatessa havaittiin, että robotin päässä sijaitsevien tuulettimien tuottama melu heikentää puheentunnistus tarkkuutta. Yleisesti ottaen tätä työtä tehtäessä saatiin ymmärrys siitä, kuinka paljon taustamelu vaikuttaa negatiivisesti puheentunnistus tarkkuuteen. Voidaan sanoa, että puheen erottaminen muusta äänisignaalista on vielä ratkaisematta puheentunnistusteknologiassa.

Googlen puheentunnistuspalvelun tarkkuutta koetettiin parantaa palveluun lähetettävän äänisignaalin käsittelyllä. Äänisignaalista yritettiin vähentää robotin tuulettimien aiheuttamaa taustamelua. Taustamelun suodattimina kokeiltiin eri taajuusalueen kaistanpäästösuodattimia sekä Fourier-analyysiin perustuvaa häiriönpoistoalgoritmia. WER-tuloksia vertailtaessa nähtiin, ettei kaistanpäästösuodattimen lisääminen äänisignaaliin vaikuttanut merkittävästi tulokseen. Sen sijaan Fourier-analyysiin perustuva häiriönpoistoalgoritmi paransi WER-tuloksia äänivirtaa palvelulle lähetettäessä, mutta se ei vaikuttanut myönteisesti tulokseen, kun palvelulle lähetettiin kokonainen äänitiedosto.

Työssä tehtyjen testien perusteella voidaan sanoa, että Googlen ulkoista puheentunnistusta ei kannata ottaa käyttöön Pepper-robotin pääasialliseksi puheentunnistukseksi, koska se ei ole tarpeeksi nopea ja sen tunnistustarkkuudessa oli myös parantamisen varaa. Kuitenkin sen kyvykkyys vapaaseen puheentunnistukseen ja siitä saadut hyödyt ovat sen verran merkittäviä, ettei sen hyödyntämistä Pepper-robotin puheentunnistamisessa kannata kokonaan sivuuttaa. Googlen ulkoista puheentunnistusta voitaisiin esimerkiksi hyödyntää Pepper-robotin sisäisen puheentunnistuksen ohella, jotta molemmista saataisiin niiden parhaat hyödyt irti.

Työstä saatuja tuloksia voidaan hyödyntää Pepper-robotin ulkoisen puheentunnistuksen jatkokehityksessä. Liittämällä ulkoinen puheentunnistus osaksi robotin ohjelmistoarkkitehtuuria, voidaan parantaa järjestelmän kokonaispuheentunnistusta ja tämän seurauksena tehdä robotin käyttäjäkokemuksesta entistä miellyttävämpi.

Lähteet

1. Häkkinen, Kaisa. 2007. Kielitieteen perusteet. Tampere: Tammer-Paino Oyy.

2. Pieraccini, Roberto. 2012. The Voice in the Machine. Massachusetts Institute of Technology.

3. D. Yu & L. Deng. Automatic Speech Recognition. 2015. Springer-Verlag London.

4. Karpagavalli S & Chandra E. 2016. A Review on Automatic Speech Recognition Architecture and Approaches. Verkkoaineisto. <https://www.researchgate.net/publication/302915903_A_Review_on_Automatic_Speech_Recognition_Architecture_and_Approaches>. Luettu 1.4.2019 : s.n.

5. Kurimo, Mikko. 2008. Puheentunnistus. Puhe ja Kieli, 15.4.2012, s. 73-83. Verkkoaineisto. <https://journal.fi/pk/article/view/5112/4616>. Luettu 19.4.2019.

6. Kurimo, Mikko. 2009. Puhuva ihminen: puhetieteiden perusteet. Helsinki: Otava.

7. Kurimo, Mikko, Puheentunnistus. Verkkoaineisto. <http://www.cis.hut.fi/Opinnot/T-61.3010/puheentunnistus_kurimo.pdf>. Luettu 19.04.2019.

8. Myers, Erin. 2017. Speech Recognition Accuracy: Past, Present, Future. Verkkoaineisto. <https://www.temi.com/blog/speech-recognition-accuracy-history/>. Luettu 21.4.2019.

9. S.Karpagavalli, R.Deepika, P.Kokila, K.Usha Rani & E.Chandra. 2011. Automatic Speech Recognition: Architecture, Methodologies and Challenges - A Review. Verkkoaineisto. <http://ijarcs.info/index.php/Ijarcs/article/viewFile/906/894>. Luettu 19.4.2019.

10. Chandra E & Karpagavalli S. 2016. A Review on Automatic Speech Recognition Architecture and Approaches. International Journal of Signal Processing, Image Processing and Pattern Recognition, 9.4.2016, s. 393-404. .

11. Huttunen, Heikki. 2003. Signaalinkäsittelyn perusteet. Verkkoaineisto. <http://cna.mamk.fi/Public/ReijoVuohelainen/DigitaalinenSignaalinkasittely/DSP-Moniste.pdf>. Luettu 19.4.2019.

12. Huttunen, Heikki. 2010. Signaalinkäsittelyn sovellukset. Tampereen teknillinen yliopisto. Verkkoaineisto. <https://docplayer.fi/38664352-Heikki-huttunen-signaalinkasittelyn-sovellukset.html>. Luettu 19.4.2019.

13. Steven W. Smith. The Scientist and Engineer's Guide to Digital Signal Processing, Chapter 14: Introduction to Digital Filters, Filter Basics. Verkkoaineisto. <http://www.dspguide.com/ch14/1.htm>. Luettu 19.4.2019.

14. Steven W. Smith. The Scientist and Engineer's Guide to Digital Signal Processing, Chapter 14: Introduction to Digital Filters, Frequency Domain Parameters. Verkkoaineisto. <http://www.dspguide.com/ch14/4.htm>. Luettu 19.4.2019.

15. Dong YuLi Deng. 2015. Automatic Speech Recognition: A Deep Learning Approach. Springer-Verlag London 2015.

16. Suvilehto, Jyry. 2017. Syvät neuroverkot. Verkkoaineisto. <https://csc.fi/es/web/blog/post/-/blogs/syvat-neuroverk-1>. Luettu 19.4.2019.

17. Johan Schalkwyk. 2019. An All-Neural On-Device Speech Recognizer. Verkkoaineisto. <https://ai.googleblog.com/2019/03/an-all-neural-on-device-speech.html>. Luettu 19.4.2019.

18. Toshiyuki S, Takayuki K,Michita I, Hiroshi I & Norihiro H. How Quickly Should Communication Robots Respond?. Verkkoaineisto. <https://dl-acm-org.ezproxy.metropolia.fi/citation.cfm?id=1349843>. Luettu 19.4.2019.

19. Pepper the Robot. Verkkoaineisto. <https://www.infosciencetoday.org/type/news/pepper-the-robot.html>. Luettu 21.3.2019.

20. Pepper. Verkkoaineisto. <https://www.softbankrobotics.com/emea/en/pepper>. Luettu 21.3.2019.

21. Robot PNG image with transparent background. Verkkoainesto. <http://pngimg.com/download/45321>. Luettu 21.3.2019.

22. Supported languages. Verkkoaineisto. <http://doc.aldebaran.com/2-5/family/pepper_technical/languages_pep.html#language-codes-pep>. Luettu 22.3.2019.

23. ALSpeechRecognition. Verkkoaineisto. <http://doc.aldebaran.com/2-5/naoqi/audio/alspeechrecognition.html>. Luettu 21.3.2019.

24. QiChat - Syntax. Verkkoaineisto. <http://doc.aldebaran.com/2-5/naoqi/interaction/dialog/dialog-syntax_full.html>. Luettu 21.3.2019.

25. Transcribing audio from streaming input. 2019. Verkkoaineisto. <https://cloud.google.com/speech-to-text/docs/streaming-recognize>. Luettu 2.5.2019.

26. Human Voice Frequency Range. 2018. Verkkoaineisto. <http://www.seaindia.in/blog/human-voice-frequency-range/>. Luettu 2.5.2019.

27. Behzad, Munir. 2012. Voice fundamentals - Human Speech Frequency. Verkkoaineisto. <http://www.uoverip.com/voice-fundamentals-human-speech-frequency>. Luettu 2.5.2019.

28. Butterworth. Aalto University wiki. Verkkoaineisto. <https://wiki.aalto.fi/download/attachments/62723056/r42_ele_kt5.pdf?version=1&modificationDate=1330290243000&api=v2>. Luettu 24.4.2019.

29. Filter Order 5. Verkkoaineisto. <https://upload.wikimedia.org/wikipedia/commons/thumb/b/bd/Filters_order5.svg/1140px-Filters_order5.svg.png>. Luettu 24.4.2019.

30. Sainburg, Tim. 2018. Noise reduction using spectral gating in python. Verkkoaineisto. <https://timsainburg.com/noise-reduction-python.html>. Luettu 24.4.2019.

31. How Audacity Noise Reduction Works. 2015. Verkkoaineisto. <https://wiki.audacityteam.org/wiki/How_Audacity_Noise_Reduction_Works>. Luettu 25.4.2019.

32. Gevirtz, Morris. 2019. The trouble with word error rate (WER). Verkkoaineisto. <https://blog.deepgram.com/the-trouble-with-wer/>. Luettu 2.5.2019.