View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Aaltodoc Publication Archive

Development of the Finnish Spoken Dialog System for an Educational Robot

Niklas Sallinen

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 23.2.2017

Thesis supervisor:

Prof. Mikko Kurimo

Thesis advisor:

M.Sc. (Tech.) Peter Smit

Aalto University
School of Electrical Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Niklas Sallinen

Title: Development of the Finnish Spoken Dialog System for an Educational Robot

Date: 23.2.2017 Language: English Number of pages: 7+40

Department of Signal Processing and Acoustics

Professorship: Speech and language processing

Supervisor: Prof. Mikko Kurimo

Advisor: M.Sc. (Tech.) Peter Smit

Spoken dialog systems are coming in the every day life, for example in the personal assistants such as Siri from Apple. However, spoken dialog systems could be used in a vast range of products. In this thesis a spoken dialog system prototype was developed to be used in an educational robot.
The main problem in an educational robot to recognize children's speech. The speech of the children varies significantly between speakers, which makes it more difficult to recognize with a single acoustic model.
The main focus of the thesis is in the speech recognition and adaptation. The acoustic model used is trained with data gathered from adults and then adapted with the data from children. The adaptation is done for each speaker separately and also as an average child adaptation. The results are compared to the commercial speech recognizer developed by Google Inc.
The experiments show that, when adapting the adult model with data from each speaker separately word error rate can be decreased from 8.1 % to 2.4 % and with the average adaptation to 3.1 %. The adaptation that was used was vocal tract length normalization (VTLN) and constrained maximum likelihood linear regression (CMLLR) combined. In comparison word error rate of the commercial product used is 7.4 %.

Keywords: Speech recognition, spoken dialog, speaker adaptation, VTLN, CM-LLR

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Niklas Sallinen

Työn nimi: Suomenkielisen puhepohjaisen dialogijärjestelmän kehitys koulutusrobottiin

Päivämäärä: 23.2.2017 Kieli: Englanti Sivumäärä: 7+40

Signaalinkäsittelyn ja akustiikan laitos

Professuuri: Puheen- ja kielenkäsittely

Työn valvoja: Prof. Mikko Kurimo

Työn ohjaaja: DI Peter Smit

Applen puhelimissa olevan assistentti Sirin tavoin puhepohjaiset dialogijärjestelmät ovat tulossa osaksi jokapäiväistä elämäämme. Puhepohjaisia dialogijärjestelmiä voi kuitenkin käyttää myös monissa muissakin sovelluksissa. Tässä diplomityössä sdialogijärjestelmän prototyyppi kehitettiin käytettäväksi koulutusrobotissa.
Suurin haaste koulutusrobotissa on lapsien automaattinen puheentunnistus. Lasten puhe on hyvin vaihtelevaa puhujien välillä, minkä takia puheentunnistus on hyvin vaikeaa yhtä akustista mallia käyttämällä.
Tämä diplomityö keskittyy pääasiassa puheentunnistukseen ja akustisen mallin adaptointiin. Akustista mallia, joka on opetettu aikuisten puheella, adaptoidaan, jotta se antaisi parempia tuloksia lasten puheen tunnistuksessa. Adaptointi tehdään kahdella tavalla: puhuja adaptointina ja keskimääräisenä lapsiadaptointina. Tuloksia verrataan Googlen kehittämään kaupalliseen puheentunnistimeen.
Kokeet osoittavat, että adaptoimalla aikusten akustista mallia puhuja kohtaisesti sanavirheprosentti (WER) saatiin laskemaan 8.1 %:sta 2.4 %:iin ja Keskimääräisellä lapsiadaptoinnilla taas 3.1 %:iin. Adaptointiin käytettiin Vocal tract length normalization (VTLN) sekä Constrained maximum likelihood linear regression (CMLLR) -tekniikoita erikseen ja yhdistettynä. Vertailukohtana käytettiin Googlen puheentunnistimen sanavirheprosenttia 7.4 %.

Avainsanat: Puheentunnistus, puhuttu dialogi, adaptointi, VTLN, CMLLR

Preface

This thesis was done at the department of Signal Processing and Acoustics in the Aalto University School of Electrical Engineering. The work was made from May 2016 to February 2017.

I want to thank Jussi Wright and A.I. Robots Oy for the interesting topic for this thesis. I would also like to thank the supervisor Mikko Kurimo and especially advisor Peter Smit for the constructive feedback and support during the thesis work.

Otaniemi, 23.2.2017

Niklas Sallinen

Contents

Abbreviations

ASR	Automatic speech recognition
CPU	Central processing unit
DFT	Discrete Fourier transform
DNN	Deep neural networks
EM	Expectation-Maximisation algorithm
G2P	Grapheme-to-phoneme conversion
GMM	Gaussian mixture model
GUI	Graphical user interface
HMM	Hidden markov model
IoT	Internet of things
LM	Language model
LVCSR	Large vocabulary continuos speech recognition
MFCC	Mel-Frequency cepstrum coefficient
OOV	Out of vocabulary
RAM	Random access memory
RPi	Raspberry Pi
RTF	Real-time factor
SD	Speaker dependent
SI	Speaker independent
TCP	Transmission control protocol
TTS	Text-to-speech synthesis
UI	User interface
VAD	Voice activity detection
WER	Word error rate

1 Introduction

Intelligent spoken dialog systems can be used in the wide range of applications. The best known applications of spoken dialog modeling are probably personal assistants such as Apple's Siri and Microsoft's Cortana. However there is many more types of applications that can take advantage of spoken dialog systems.

Internet of things (IoT) is rapidly developing technology and spoken dialog systems could take it even further. It's basic idea is to connect every machine to each other via internet. By combining that idea to the talking robot or spoken interface it is possible to develop new kind of natural ways to interact with different machines through a single interface.

The interaction with machines via IoT is also in some products combined with a personal assistant. Thus, the end product is a human like assistant, that can make various different things and help people in daily tasks, for example reminding to take the umbrella if it is raining.

Another application where utilising robots could help is in the education. By using robot most of the mechanical work related to revising exams and exercises can be automated and the teacher can get data that is already in the form that describes the knowledge that students have gained. It can save time for teacher, which can be used to teach the things that are difficult in more detail. The most difficult topics can also be detected easily by tracking the progression of each student. Thus, teaching can be more personalized for each group.

1.1 Description of the Project

The goal of the project is to design a robot that is used in teaching and examinations in schools. The robot is supposed to give new ways to get information about the improvement of the students' skills via informal spoken examinations.

The progression of children will be followed multiple times in a week, which makes the information about knowledge of the children updated often. Thus, it is faster to realise than with conventional exams, if someone is falling behind or progressing so much that more difficult exercises are needed. With the robot, that is possible without using a significant amount of the teacher's working hours.

This thesis describes the development of the spoken dialog system and its different sub-parts. The spoken dialog system described in this thesis is designed for Finnish language as the product will be used in Finland at the first stage. The dialog system consists of automatic speech recognition system (ASR), text dialog modelling and text-to-speech (TTS) speech synthesis. The other components such as movement and vision are left out of this thesis. The main focus of the thesis is in the ASR part. The block diagram of the system developed in this thesis is presented in Figure 1.

In the project a ready-made systems are used to reduce the development time of the prototype. When using third-party developed systems the price of the usage is important as well as the performance. However, the evaluation of the prices is left outside of this thesis. The preliminary platform for the spoken dialog system is a Raspberry Pi 3B, which is presented in Section 6.2.1.

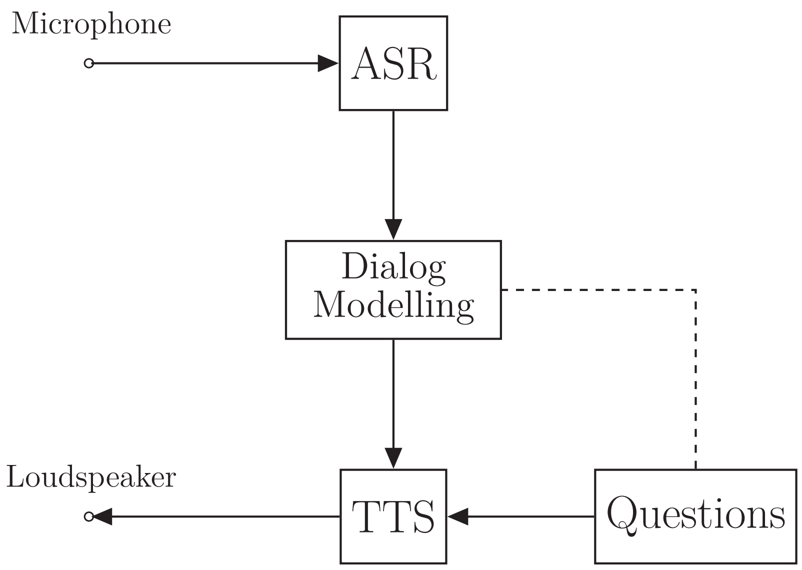


Figure 1: Block diagram of the spoken dialog system.

1.2 Requirements for the System

As the robot is planned to be used mainly in schools, the main concern in speech recognition is that is should work well with voices of children. The error rate of recognition should be almost perfect for the keywords that are in this application, for example digits in the math applications. For other content it is acceptable to get worse accuracy.

In this thesis the improvement in the accuracy of the speech recognition is researched, when speaker or average child adaptation adaptation is applied to the adult model and test speakers are children. Similar research has been done in [1, 2, 3] with different languages, but not with the same adaptation techniques. In this thesis similar results goal is to produce similar results with Finnish language and vocal tract length normalization (VTLN) and constrained maximum likelihood linear regression (CMLLR).

Requirements for the processing speed for each part depends on the speeds of the other parts. The time reduced in one part will increase the time that can be used in others. In human-to-human interactions the gap between speakers varies most commonly between 200 ms and 700 ms [4, 5, 6]. As the spoken dialog with the robot should feel like natural and fluent, the gap from the end of the users speech to the beginning of the reply of the robot should be similar to values presented. It is worth to be noted that not only too long gaps feel unnatural but too short gaps do not feel natural either. However, the speed of the ASR was only measured as other sub-parts take so much less time and the impact of those to the processing time is small.

In speech synthesis the main task is to optimise the speed of the processing and get the synthesis to work in real-time. The advantage of the real-time processing is that the selection of the sentences and answers can be more spontaneous, when they don't have to be completely defined previously. The other benefit is lower storage usage, which is important when using such a low-end hardware as a Raspberry Pi.

Even if the speed of synthesis is the most important factor, the quality of the speech should be as good as possible. The minimum requirement is that speech is intelligible. However, it is preferable that it is as natural as it can be when the speed is optimized. However, in this thesis the evaluation of the system was left out as the main scope of the thesis is in the speech recognition. Thus, the speech synthesis was chosen to get satisfactory results with the prototype.

The methods and measures used for the evaluations are presented in Section 7.3.

1.3 Structure of the Thesis

The basic theory about the automatic speech recognition is covered in Chapter 2. The adaptation methods used in the experiments are presented in the same chapter.

Dialog modelling and natural understanding methods that are implemented into the prototype are presented in Section 3. The basic theory of the TTS is given as a background information in Chapter 4.

Chapter 5 presents the ASR systems compared in the experiments as well as the models and other details related to the speech recognition systems. Also the TTS system used is briefly discussed.

The implementation of the prototype and also details of the hardware are presented in Section 6. The user interface and the basic functioning of the prototype are explained.

Chapters 7 and 8 describe the experiments and the evaluation measures and present the results.

2 Automatic Speech Recognition

Automatic speech recognition is a combination of different technologies from different fields, such as machine learning, linguistics, and acoustics. The goal of an ASR system is to automatically generate text from a speech audio. Large vocabulary continuos speech recognition (LVCSR) is a subfield of ASR, where the speech is assumed to contain a wide range of different topics and the recognizer should work generally for any of those.

This section describes the fundamental concepts of an ASR system consist of Gaussian mixture models (GMM) and Hidden Markov models (HMM). The section will also include the most important design factors for speech recognition. Figure 2 presents the most common structure of an automatic speech recognition system. All of the blocks in the diagram are explained in detail in this chapter.

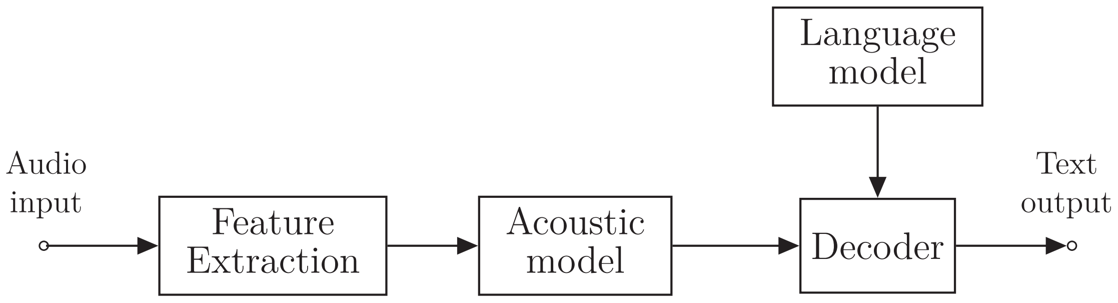


Figure 2: Block diagram of an automatic speech recognition system.

2.1 Feature Extraction

The purpose of feature extraction is to change a representation of a digital audio signal into a sequence of feature vectors. Each vector contain information, about one frame of the audio signal. The best parametric representation for the speech recognition is Mel-Frequency cepstrum coefficients (MFCC) [7], which are presented in this section.

In the first stage of the feature extraction the high frequencies are boosted with a pre-emphasis filter. After filtering, the signal is divided in small windows inside of which the signal is assumed to be stationary [8]. Then the spectral features of each window is extracted with discrete Fourier transform (DFT):

$$X(k) = \sum_{n=0}^{N-1} x[n]e^{-j2\frac{\pi}{N}kn}, \tag{1}$$

where $X(k)$ is the signal transformed to the frequency domain, $x[n]$ is the discrete signal in the time domain and N is the length of the window. [8]

The Mel-frequency scale is non-linear and it mimics the behaviour of the human auditory system. At the low frequencies a human ear is more sensitive to the changes in the pitch than at the high frequencies, which is taken account in the Mel-frequencies. The results of the Fourier transform are warped to the Mel-frequencies using triangular bandpass filters, that are equally located along the Mel-scale and then the logarithmic energy of each output of the filters is computed. A Mel-filterbank is illustrated in Figure 3. Mel-frequencies can computed with following equation:

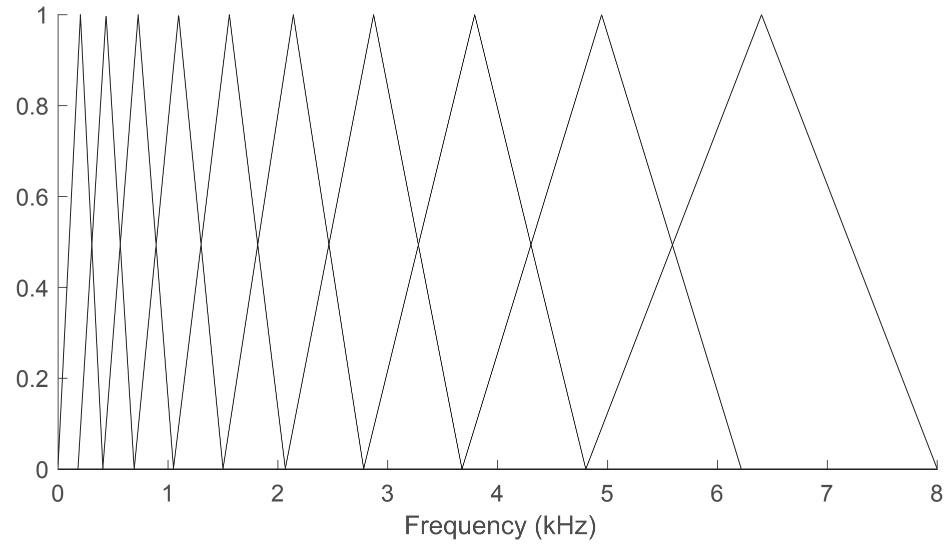


Figure 3: Ten band Mel-filterbank.

$$mel(f) = 1127 \ln\left(1 + \frac{f}{700}\right) , \tag{2}$$

where f is the original frequency in Hertz scale. [8]

After computing the logarithmic Mel-spectrum, the final step is to calculate a cepstrum of the spectrum. The Mel-frequency cepstrum is a discrete cosine transform of the logarithmic energies calculated in the earlier stage. The discrete cosine transform is calculated with Equation 3.

$$c[n] = \sum_{m=0}^{M-1} S[m] \cos\left(\frac{\pi n(m+\frac{1}{2})}{M}\right), \tag{3}$$

where $S[m]$ is the output logarithmic energy of the Mel-filterbank and M is the amount of the bands in the filterbank. [9]

2.2 Acoustical Modeling

The acoustic model contains the probabilities of different audio signals for each phoneme. It is usually implemented with a hidden Markov model (HMM). In the HMM data, there is a three state model for each set of three sequential phones —triphone— that occur in the training data. In addition to those there are some some special HMMs , such as silences of different lengths.

The HMMs used in the acoustic model are mainly three state HMMs. The structure of the model is shown in Figure 4. The model includes transition probabilities a_{ij}, whose are independent from the observations. That is probability of going from state i to state j. The model contains as well probability distributions for the observations. The observation probabilities are presented as Gaussian mixture models (GMM), whose are used to calculate the probability for each feature vector.

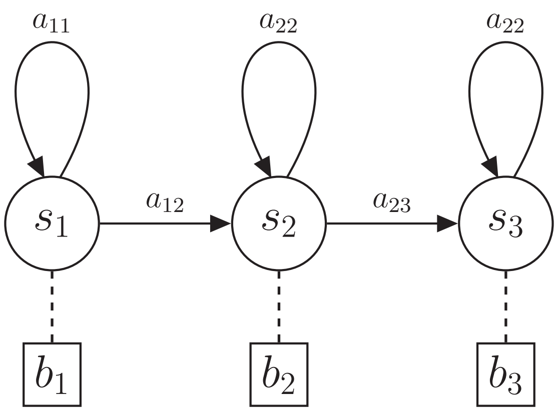


Figure 4: Three state HMM. a_{ij} are transition probabilities from state i to j and b_i are observation probability distributions for corresponding states.

A Gaussian mixture model is a weighted summation of Gaussian distributions. By combining finite number of components, any distribution can be formed with the GMM defined as the following equation

$$f(x|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{M} w_k \mathcal{N}(\mu_k, \Sigma_k) \tag{4}$$

$$f(x|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{M} w_k \frac{1}{\sqrt{2\pi|\Sigma_k|}} \exp\left[(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)\right], \tag{5}$$

where $\mathcal{N}(\mu_k, \Sigma_k)$ is probability density of the multivariate normal distribution with a mean of μ_k and a covariance of Σ_k, w_k is a weight for k:th distribution and $\sum_{k=1}^{M} w_k = 1$. Figure 5 shows how the separate distributions are combined together. In the left subfigure the distributions are shown separately and on the right side the sum of those is presented. [8]

2.3 Language Modeling

The language model (LM) defines the probabilities for each sentence, without taking into account the audio signal input. It contains the information to calculate the probabilities for all the different sentences that can be formed. The language model is trained using a corpus of text and the probabilities of the model correspond to the data used for training. The very basic model used for language is an n-gram model.

The n-gram model gives probabilities for each combination of up to n words found in the lexicon. The conditional probabilities for the combination of the words are calculated by counting the occurances of the combination of the words in the

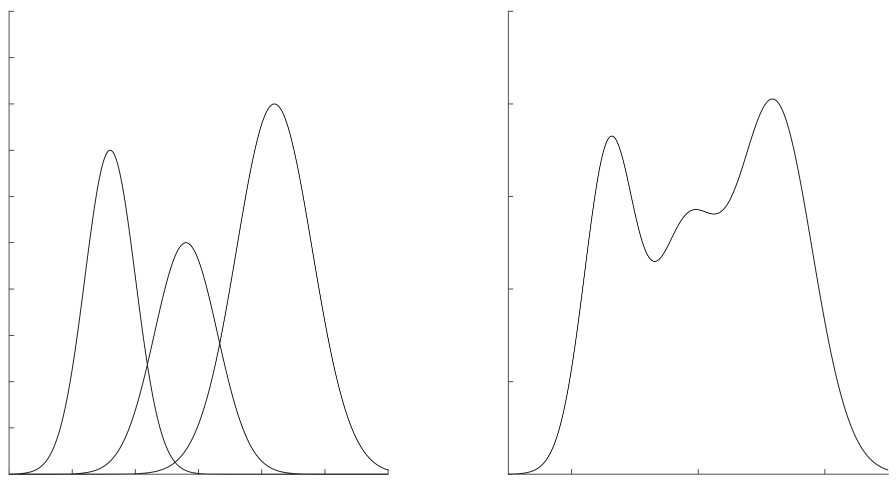


Figure 5: Illustration how a GMM is constructed from separate Gaussian distributions. On the right the gaussian distributions in the left figure are combined together as a GMM.

training corpus and dividing that with the count of the previous words in the corpus. Equation 6 shows an example of that for 2-gram (bigram) and Equation 7 for 3-gram (trigram).

$$P(w_i|w_j) = \frac{C(w_i, w_j)}{C(w_j)}, \tag{6}$$

$$P(w_i|w_j, w_k) = \frac{C(w_i, w_j, w_k)}{C(w_j, w_k)}, \tag{7}$$

where w_i, w_j and w_k are words for whose the probability is calculated and $C(w)$ is the count of the word w in the corpus. For the sentence the probability is the product of all the n-gram probabilities found in the sentence. The example of calculating the probability for the sentence with bigrams is shown below. [9]

$$\begin{gathered} P(\text{A boy bought an ice cream}) = \\ P(\text{A}|\text{<s>})P(\text{boy}|\text{A})P(\text{bought}|\text{boy})P(\text{an}|\text{bought}) \qquad (8) \\ P(\text{ice}|\text{an})P(\text{cream}|\text{ice})P(\text{</s>}|\text{cream})\ ,\ [9]. \end{gathered}$$

In the calculations the interpolated Kneser-Ney smoothing is used to handle the zero probabilities. This will give some non-zero probability to the combinations that are not found in the corpus, and the method is based on the assumption that the corpus is not perfect. For example the probabilities for the bigrams are calculated with Equation 9 and unigrams with Equation 11.

$$P(w_i|w_j) = \frac{\max(0, C(w_i, w_j) - D)}{C(w_j)} + \lambda(w_j)P(w_i), \tag{9}$$

where

$$\lambda(w_j) = \frac{D}{C(w_j)} \quad |\{w : c(w_j, w) > 0\}|, \tag{10}$$

and $C(w_j)$ is the count of the context words w_j, $C(w_i, w_j)$ is the count of the words w_i and w_j appearing together, D is discount weight, and $|\{w : c(w_j, w) > 0\}|$ is a number of different words appearing with w_j. $P(w_i)$ is defined as below in Equation 11.

$$P(w_i) = \frac{C(w_i)}{C(w_{all})} .[9] \tag{11}$$

To improve the accuracy of the speech recognition, sub-word units known as morphs can be used in the language model instead of words. This approach decreases the vocabulary size needed for the similar OOV word percentage. In Finnish compound words are written together and meaning of the word can be modified by adding suffixes to it. Thus, it is more effective to divide the words in smaller units that have certain meaning than list all the different inflections. [10]

2.4 Language Model Lookahead

A language model lookahead is similar to the actual language model, but it is smaller and thus less demanding computationally. The lookahead is used to calculate initial probabilities and based on these probabilities the actual language model is applied only for the most probable utterances to save computing effort. [11]

2.5 Lexicon

A lexicon is a list of all words in the language model. It contains information how each word is pronounced. The information of the pronunciation is given as a sequence of phones or more often triphones. Triphones are used in this thesis. The lexicon is the model that binds a language model and an acoustic model together. In practice the lexicon is a list of words combined with sequence of the triphones, whose store the pronunciation information. Example of the word-pronunciation pair is shown in Figure 6

```
yksi   _-y+k  y-k+s  k-s+i  s-i+_
kaksi  _-k+a  k-a+k  a-k+s  k-s+i  s-i+_
kolme  _-k+o  k-o+l  o-l+m  l-m+e  m-e+_
```

Figure 6: Example of three word-triphone pair in the lexicon.

2.6 Decoder

A decoder is a sub part of the speech recognition system, that transforms the outputs from the models to the sequence of words, that is most probable for the sound input.

The most used method for decoding HMMs is an algorithm known as the Viterbi algorithm. [9]

The algorithm defines probabilities for each path, the sequence of states from the beginning of the model to the end. The calculation of the different path probabilities starts with initialising the probabilities. For every state there is an initial probability defined. The initial probability is then multiplied with the probability of the observation X in that specific state.[8, 9]

After the probabilities are initialised the probabilities of the next state of the path are calculated recursively using the probabilities of the previous states, transition probabilities between the states and the observation probabilities in the next state. Those three are multiplied together and the probability of the next state will be the probability of the incoming path that has the highest value. The calculations of the state probabilities are shown in the equations below:

$$V_1(i) = \pi_i b_i(X_1), \tag{12}$$

$$V_t(j) = \max_i \left(V_{t-1} a_{ij}\right) b_j(X_t),\ 1 < t < T \tag{13}$$

where π_i and b_i are the initial and the observation probabilities of the state s_i respectively. a_{ij} is the transition probability between the states s_i and s_j and X_t is the observation at the time index t. [8, 9] The basic idea of this step is visualised in Figure 7.

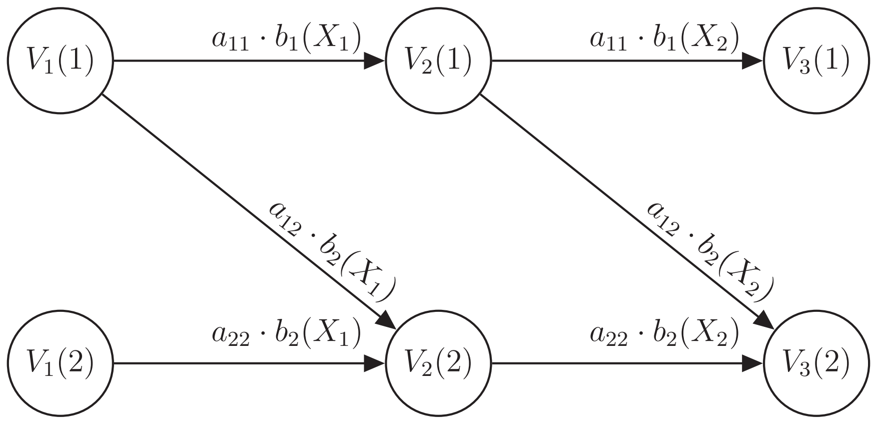


Figure 7: The illustration of the Viterbi search for a two state HMM, where a_ij are the transition probabilities, b_j are the observation probabilities for each observation X_t at the state s_j. $V_t(j)$ is the score of the state s_j at the time t.

The different path probabilities are calculated until there is no more observations. Then from the probabilities of the last states the maximum is found and that is the score of the best path. After finding the score of the path the actual path has to be traced. That is done by going through each step of the path backwards and in every step finding the highest scored state in the previous time index until the initial state is reached. [8, 9]

In LVCSR the number of the probabilities for different possible paths is large. Thus, some optimization has to be done to get the speed of the recognition as high as possible without sacrificing the performance significantly. The two main techniques used commonly are pruning and sharing. [9]

Pruning means that the decoder keeps calculating only some predefined amount of the most possible paths and throws away the last promising paths. Sharing is an application of dynamic programming. The values of intermediate paths are saved and used later for the paths that use the same intermediate path to avoid the calculations of the same paths multiple times. [9]

After the acoustic model scores are decoded to the sequences of states, the intermediate transcriptions are generated from the sequences with the help of lexicon. After that the intermediate transcriptions are evaluated with the language model in addition to the sequence scores. The probabilities for the transcriptions are calculated as shown in Section 2.3. [9]

The language and the acoustic models are in the different probability spaces. To overcome this problem the language model weight is applied to the probability of the language model to balance the probabilities by adusting the probability of LM. This is done by Equation 14.

$$P(W)^{LW}, \tag{14}$$

where LW is the language model weight which is defined empirically to maximize the performance of the recognizer and usually $LW > 1$. [9]

2.7 Voice Activity Detection

Voice activity detection (VAD) is a really important part of the ASR system. It is the technique, which decides when the input signal is containing speech. This information is used to omit the parts that do not include speech, so they are not given to the recognizer to save computational capacity for actual recognition. The ends of utterances are detected as well with VAD and when it is detected the recognizer can give its final results.

2.8 Adaptation

The models trained with the data that contain multiple speakers, aim to model the average speech and thus, they are referred as a speaker independent models (SI). However, the accuracy of the recognition is better, if the model is concentrating only on a one speaker, which means it is speaker dependant (SD).

Speaker adaptation is a method which targets to match the general SI model to certain characteristics of an individual speaker. This is especially useful if the users voice differs much from the speakers from whom the data is used to train the actual model. In this application the general adult model is supposed to be adapted to perform well with children.

Children's speech recognition has multiple difficulties. The most apparent difficulty is that the speech of the children varies between speakers much more than with adults. The differences occur mainly because the vocal folds and tract are still developing and because children are still learning the pronunciations of words. With adults these properties are much more static between speakers. Also, collecting data from children could be difficult as children usually does not stay still and concentrated long enough to record large data sets. [12]

Adaptation can be done different ways. The model can be adapted to some characteristics of acoustics or to an individual speaker. In this thesis the adaptation is done to the characteristics of average children speech as well as for the individual speakers. The basic idea behind the adaptation is to transform the observation probability distributions of the acoustic model to match better the adaptation data. In practice this means updating the means and covariances of the Gaussians in the model. When the general speaker independent model already exists less data is needed to adapt the model than training a new SD model with such a characteristics. [13]

Training the adaptated model can be done two ways: supervised and unsupervised. Supervised training need annotated data, which includes audio files and transcriptions of their content. The correct transcriptions are used to align the phonemes in the model with the adaptation data. In unsupervised training only the audio is available and the transcriptions used for the adaptation are acquired by using the recognizer with the initial model and using the recognized transcripts for the adaptation. In practice this means that unsupervised adaptation can be applied during the recognition and supervised has to be done before use.

Three most used adaptation techniques are Vocal Tract Length Normalization (VTLN), Constrained Maximum Likelihood Linear Regression (CMLLR) and Maximum a Posteriori (MAP) adaptations. In the next sections the basics of them are presented.

2.8.1 VTLN

In the VTLN the basic concept is to linearly warp the frequencies of the input speech signal. The warping models the difference in length of the vocal tract between the average speaker in the training data and the speaker in the adaptation data. This is done by measuring the ratio between two vocal tract lengths and then the ratio is used to calculate the corresponding warping factor. [14]

Normalizing the length of the vocal tract is a good approach to adapt between the speakers that has different sizes of vocal tracts. The tract with same shape, but different length has similar frequency response, but it is shifted in the frequency domain. That works rather well for example between women, men and children. [14]

The warping factor is usually estimated by maximizing the likelihood of the observation by trying systematically different values for the warping factor α. Equation 15 presents the training of warping factor..

$$\hat{\alpha}_{ML} = \underset{\alpha}{\operatorname{argmax}}\, p(\boldsymbol{X}^{\alpha}|\lambda, \boldsymbol{W}), \tag{15}$$

where $\boldsymbol{X}^{\alpha}$ is features warped with the factor α and λ is the acoustic model and $\boldsymbol{W}$ is the transcription from the recognizer or the annotated data. [15]

The VTLN can performed as a feature adaptation, which means that the feature vectors are transformed, rather than the actual acoustic model. The adapted features are calculated with Equation 16.

$$\boldsymbol{X}^{\alpha} = \boldsymbol{A}^{\alpha}\boldsymbol{X}, \tag{16}$$

where $\boldsymbol{A}^{\alpha}$ is a linear transform matrix that can be calculated beforehand and $\boldsymbol{X}$ contains the original features. [16]

2.8.2 CMLLR

In CMLLR adaptation the parameters of the Gaussians are updated with a linear transformation. Multiple or even all of the Gaussians could be transformed with a single regression matrix and a bias vector. Thus the adaptation can be done with a small amount of data.

The updates are done with Equations 17 and 18:

$$\hat{\mu} = \boldsymbol{A}\mu + b, \tag{17}$$

$$\hat{\Sigma} = \boldsymbol{A}^{T}\Sigma\boldsymbol{A}, \tag{18}$$

where $\boldsymbol{A}$ is the estimated regression matrix, b is the bias vector and μ and Σ are the original mean and the covariance respectively. The regression matrix $\boldsymbol{A}$ is estimated with the Expectation-Maximisation (EM) algorithm, so that the $\boldsymbol{A}$ maximises the likelihood of the adaptation data. That means choosing the transform matrix so that the probabilities of the correct phoneme sequences are maximised. [13]

With Maximum Likelihood Linear Regression only the feature vectors can be adapted as well as with the VTLN. Then the acoustic model can be used as it is. Thus the implementation of the adaptation in much easier than adapting the acoustic model. If the feature vectors are transformed the procedure is made as follows:

$$\hat{x}_t = \boldsymbol{A}^{-1}x_t + \boldsymbol{A}_c^{-1}b, \tag{19}$$

where x_t is the original feature vector. [13]

2.8.3 MAP

In MAP adaptation every Gaussian is updated separately. Thus, the adaptation data has to contain most of the triphones that are in the acoustic model. This is the requirement as only those probability distributions are updated that are included in the adaptation data.

Maximum a Posteriori estimation is found with the formula:

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\ f(x|\theta)g(\theta), \tag{20}$$

where $f(x|\theta)$ is likelihood of x and $g(\theta)$ is a priori distribution of θ. The priori distribution in the adaptation is got from the original general model. The MAP estimate is a weighted mean of the ML-estimation of the adaptation data and the priori distribution. [9]

The means are estimated with the following equation:

$$\hat{\mu} = \frac{\tau\mu_0 + \sum_{t=1}^{T}\gamma(t)o_t}{\tau + \sum_{t=1}^{T}\gamma(t)}, \tag{21}$$

where o_t is an adaptation vector, μ_0 is a priori mean, τ is a meta-parameter and $\gamma(t)$ is the probability of the Gaussian at the time t. The meta-parameter τ is used for weighting the adaptation between the ML estimation of the mean of the adaptation data and and the priori mean. [13]

3 Dialog Modeling

The dialog modelling is the part of the system that gets the transcription from the speech recognizer and chooses or generates the answer text and sends it to the speech synthesiser. The response is chosen based on the content of the transcription.

The basic idea of the dialog model for the robot is that it can recognize people and greet them by asking the name of the user. After that it will start asking questions and evaluate if the answers are correct and give spoken feedback according to the evaluation. The flow chart of such a dialog system is shown in Figure 8.

In this section two sub parts of the dialog model used are discussed in detail. Natural language understanding evaluates the content of incoming transcription and and gives information for the answer generation, about the content of the input from the user. In the latter section the model for the answer generation is presented.

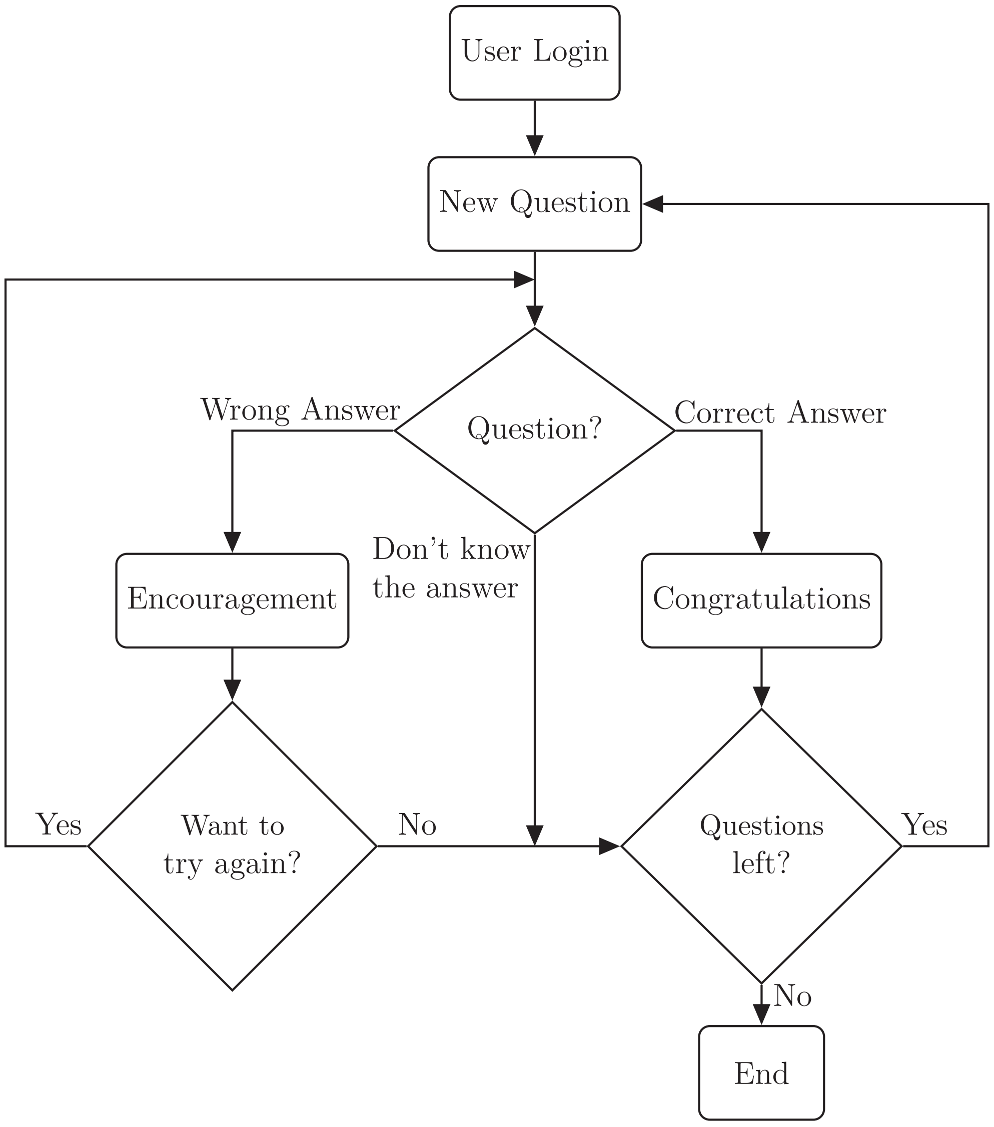


Figure 8: Flow chart of the dialog system.

3.1 Natural Language Understanding

The evaluation of the answers is made using keyword spotting. Keyword spotting means detecting selected words from a speech. The detection can be done with help of LVCSR system, which outputs lattice of phonemes or transcriptions. [17]

In this application the keyword spotting is implemented by searching the selected words from the single text transcription or the from the n best results. This approach is chosen as it gives freedom to use any kind of a speech-to-text recognizer and thus gives more options for the recognition. The choice makes adding new items to the selected words easier and later phase same recognizer could be used also for more complex tasks, with out changing the whole system. Also the recognizer could be replaced easily if needed. In this application the benefits overcome the worse performance compared to the other options. [17]

The other choice could have been searching the keywords using for example the phoneme probabilities. With this approach the accuracy of the keyword spotting can be improved significantly. However, this approach would limit the choices of ASR system to the custom built system as most of the commercial systems don't give the phoneme probabilities. [17]

In addition to only searching the words from the the text the confidence of the transcription, given by the recognizer is used. The confidence in the context of ASR is a measure estimating the reliability of the recognition result [18]. Using the measure helps to increase the performance of the system as it gives more information about different transcriptions, that can used for the evaluation.[19]

The confidence of the transcription is at the first place used to prioritise the different transcription alternatives that include keywords. The confidence of the transcription must be higher than defined threshold value to be processed at all.

The selected keywords are the most common forms of correct and false answers and some commands for the robot, including moving to the main menu and skipping the question. Each word in the list is bound to the tag, which informs the system how to proceed if the word with the tag is found from the speech.

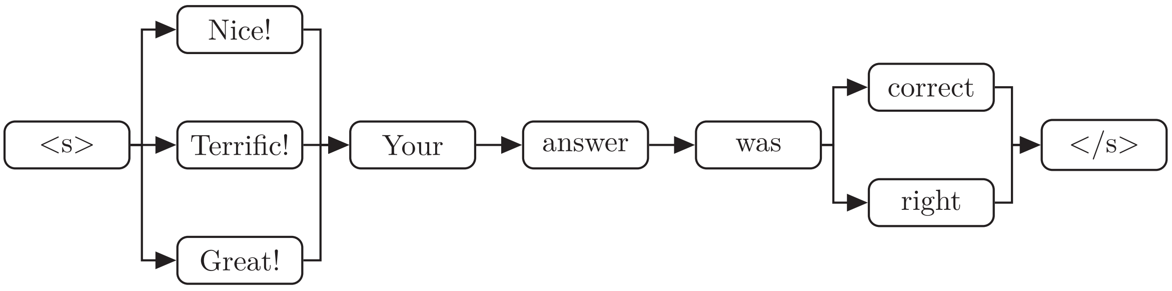
3.2 Response Generation

The different sentences that robot can say are stored in the file. All of the sentences have similar tags than the keywords presented in previous section. The tags are values that tell, if the keyword is the correct answer, wrong answer, or some other keyword. The correct answers are marked with tag value 1 and wrong answers with value 0 for example.

The tag that the keyword found from the transcription defines the tag used to search the reply for the feedback. If no selected words are found the initial tag is used. The initial tag is defined depending on the type of the question. If the question is a multiple choice question the question is repeated. Generally in the questions the initial tag is similar to the tag for the incorrect answer.

The text for the speech synthesis is chosen randomly from a defined list of different kind of grammar networks that have the same tag that was detected. The finite state grammar networks contain information which kind of sentences are possible to form.

The finite state grammar network defines every sentence that is possible to generate. In Figure 9 the example of a simple network is shown with all the different sentences that can be generated with it. Each path in the network generates a sentence that can be used as a reply. [20] The path is chosen randomly and in this application each path from the start of the sentence <s> to the end of the sentence </s> have a same weight.



```
<s> Nice! your answer was correct </s>
<s> Nice! your answer was right </s>
<s> Terrific! your answer was correct </s>
<s> Terrific! your answer was right </s>
<s> Great! your answer was correct </s>
<s> Great! your answer was right </s>
```

Figure 9: Grammar network example and the sentences that can be generated from that.

The replies are utterances that congratulate for the correct answers and encourage to try harder if the answer is wrong. There is also other replies for multiple different cases. Randomness gives variety to the spoken utterances and decreases the repetitiveness, which could be annoying.

The questions for the robot are stored in the database and are stationary, but there are some parameters that are randomized to get variety to them. This means for example in the math questions the structure of the question is static but the numbers in them are picked randomly. Also when the system talks to the user it can call the user by the name.

4 Text-to-Speech Synthesis

In this section the basic principles of text-to-speech synthesis (TTS) are described. This section contains techniques used in the text-to-speech synthesis in general, not the precise methods used in the system chosen. The purpose of the explanation is to give to the reader the basic knowledge of the speech synthesis.

TTS is a speech processing area, which is concentrating to artificially generating speech audio from written text. In the first phase the input text is parsed to separate the sentences and then the sentences are split in to words. After the words are extracted the words are encoded to the sequence of phonemes, which gives some information how the word is pronounced. After the pronunciation is defined the phonemes are converted to audio signal. In Figure 10 the generic structure of the TTS system is presented. [21]

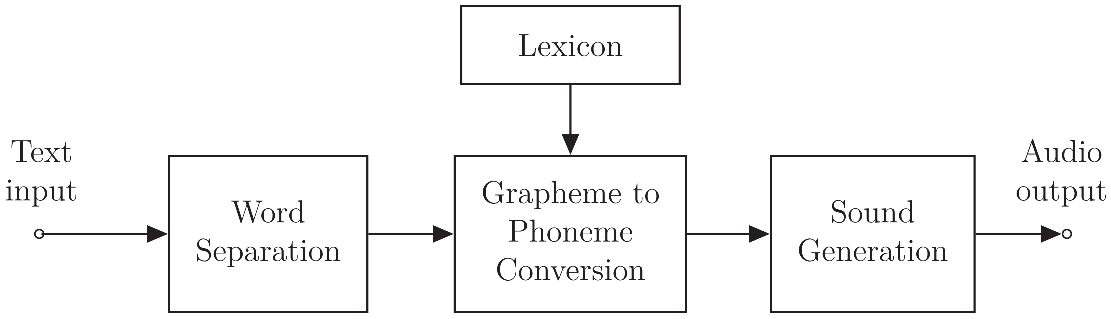


Figure 10: Block diagram of a generic simplified text-to-speech system.

4.1 Words to Phonemes

The simplest way to generate the sequence of the phonemes from the words is to have lexicon similar to the one used in speech recognition presented in Section 2.5. This is the model which has the pronunciation rules for all the words stored. However, this approach arises a problem when the word that is given as an input is not included in the lexicon. Thus, there have to be used some more advanced grapheme-to-phoneme (G2P) techniques in addition to the lexicon to get all the words handled. [21]

The simplest solution is to make pronunciation rules for generating a phoneme sequence from the text —the graphemes—. This is known as a rule-based grapheme-to-phoneme conversion. The main drawback of rule-based G2P is that the definition of the rules might be difficult, depending how complicated the language is. For the good coverage of the synthesizer the designer of the rules needs to have specific linguistic skills. The other draw back is the irregularity of the language, but the most frequent exceptions can be handled with the dictionary. However, in Finnish the words are pronounced as they are written so the task is rather simple. [22]

The other approach to make G2P conversion is a data-driven method. The method is based on the machine learning. Thus, the conversion is done by training a model with large amount of the data containing different grapheme-to-phoneme conversions. From the data system will learn how the graphemes are converted to phonemes. This approach is easier to use as it does require linguistic skills for the trainer. However, also this approach has similar problem with the exceptions of the language, as the model is basically generalisation of the pronunciation rules. In this thesis the detailed analysis of the method is not done. [22]

4.2 Sound Generation

The simplest method to convert the phoneme sequence in to the actual audio signal is the source filter model. The method is based on the human speech production. The source mimics how glottis creates for voiced phonemes the pulse wave forms, which have wide frequency content. For the unvoiced phonemes the source sound is usually noise, which frequency content is dependent on the phoneme that is modeled. Figure 11 shows the illustration of the model. [21]

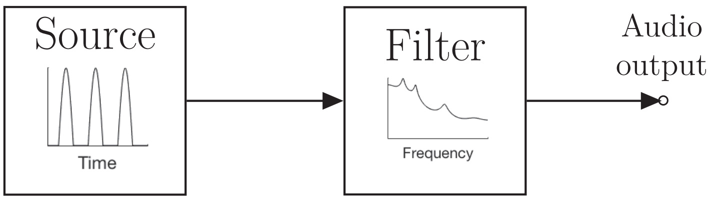


Figure 11: The block diagram of a source-filter model.

Then the signal from the source is filtered with filters that are designed to match the average frequency response of the vocal tract of a human. The frequency response is different with each phoneme as the different phonemes are produced by changing the shape of the vocal tract. [21]

5 Systems Used

In the following subsections two different ASR systems are presented. Those two are Google's Cloud Speech API and Aalto ASR, developed in Aalto University. The details that are unique to each option and the performances are the main concepts of the subsections. The TTS system used is described shortly as well.

5.1 Aalto ASR

Aalto ASR [23, 24] is a C++ application developed by the speech group of Aalto University. The application contains functions for building automatic speech recognition system and some complete implementations as well. Since AaltoASR does not provide complete models, unlike commercial systems, the acoustic and language models have to be trained by the user or acquired separately. The models used are presented in the following sections.

In this application Aalto ASR is chosen to be used in experiments, because it is developed for primarily Finnish language and it has acoustical models in Finnish that are available for use in this master's thesis project. The availability of the models in Finnish was the main decision factor as the end product will be piloted in Finland for children.

One major benefit of using a system, which is not trained, is the ability to define the language model. In the specific recognition tasks it is very useful to get to define the set of words the could be recognized. However, this approach has also disadvantage. When the size of the dictionary is reduced, the probability of out-of-vocabulary (OOV) words increases. Although, in the examination task it is fair to assume that most common answers are known and thus included in the language model.

The other advantage of Aalto ASR is that it supports speaker adaptation. When most of the users are children and generally acoustical models are trained with adult speech data the recognizer will not work well. Usually the most of the models and complete systems are designed to be used with an adult voice. This can be overcome with speaker adaptation. This is described in more detail in Section 2.8.

The major disadvantage of the speech recognition system is that it do not include voice activity detection (VAD). Voice activity detection is a really important feature as it is used to decide if the speech has ended. That is important as the recognizer will only return the results when the audio stream has ended. This is done to simplify the communication protocol [25]. Thus the voice activity detection has to be implemented separately for the system, before it can be used in the application presented in this thesis.

5.1.1 Acoustic Model

The acoustic model used with Aalto ASR was trained using a corpus of speech in Finnish from Speecon project [26].The same model is used as well in [25].

The Finnish speech data from the Speecon project consists of 600 speakers. 550 of them are adults and 50 are children. In the acoustic model only 510 of speakers were included and all of them were adults. 50 % of the speakers were male and the other 50 % were female. In the training the amount of speech data used was about 63 hours and 28,072 utterances, including sentences, word sequences, and individual words. [25, 26]

The speech was recorded using four different microphones, each in different position. There were two close distance microphones one was positioned similarly to the hands free and the other was headset microphone. The third microphone were at the distance of 0.5–1 m and the fourth was in the far-field. In this model only the recordings from the close distance mcrophones and 0.5–1 m from each speaker was used. The speech is recorded using 16 kHz sampling frequency and quantised with signed 16-bit encoding. The recordings are in raw format. [25, 26]

To get any decent results the audio signal to be recognized should be captured with same sampling frequency, same format and using similar codec or transformed to have similar properties before the recognition. It is worth to be noted that there are several uncompressed raw formats that could all be used if the other parameters are the same.

The model was trained using discriminative training scheme. The initialization of the model parameters was made with the maximum likelihood estimation. The final optimization of the parameters was made with the maximum mutual information estimation using word lattices.[25, 27]

5.1.2 Language Model

The language model used with the Aalto ASR is a 6-gram morph model "morph40000". The vocabulary size of the model is 8429 morphs. The model is trained with text data from Finnish news papers. The morph model suits better than a word model to this application, as the text data available is very different from the use case. Thus with the sub words the coverage of the words is better, especially with a morphologically rich language like Finnish.

Also the lookahead model was used in the Aalto ASR in the project. The model was trained with same data than the actual language model but instead of the 6-gram model the lookahead was a bigram model.

5.1.3 Adptation

As the acoustic model for the speech recognition was trained with the data that consist of only adult speech, and the end users will be mainly children the error rate will most probably be quite high. This is due to the fact that the speech of children differs many ways from the adult speech. To decrease the error rate, the model can be adapted with the speech data from children.

The adaptation in the applications is made using VTLN and CMLLR methods separately and combined together. Combining the methods improves performance even more. The adaptation is applied as a speaker adaptation and as an average child adaptation. The average child adaptation can be applied even for the speaker for whom there is no personal adaptation data available. [28]

5.1.4 Server for Aalto ASR

Server for the Aalto ASR is based on the same server used in [25].

The server backend is an executable that is called when the client is connected to the server via tcp port. The server receives audio packages, that are streamed from the microphone, from the client. The processing of the packets is started immediately when the first packed has arrived. The results are returned when the recognizer has reached the end of the audio stream sent.

5.2 Google Cloud Speech API

Google Cloud Speech API [29] is an automatic speech recognition system based on deep neural networks (DNN) and it is developed by Google Inc. The system is in the beta testing phase at the time of this thesis. The recognizer supports 88 languages including Finnish. Due to this it is quite simple to change the language of the recognition if needed later. However, all the other parts should be translated as well, which might need rather large amount of work.

In comparison to the previously presented Aalto ASR system this system is already trained and ready to use. Meaning that the user is not allowed to change the details in the model or parameters used. That can lead to worse results because the models and parameters are not optimized for the task. However, a user can give to the system a set of words that give a context for the recognition. This is especially useful if the speech in the usecase contains names.

Even though, the recognizer is a very closed system it can perform well in this application as well, if in the training of the system is done with the large amount of data and the methods used are chosen well. In this recognizer that is likely so it is worth to be tested if it outperforms Aalto ASR in this task.

The recognizer can return n-best results. With the multiple results in the processing phase the results that are not relevant in the context can be ignored. This helps to compensate the problem resulting from the models that cannot be changed.

The benefit of the pre-trained system is that it is much easier to use and faster to start to use. Google also provides the libraries for the front-end which decreases time needed for the implementation of the interface. The front-end libraries are written in Python, Java and Node.js programming languages and with small effort it is possible to build a working speech recognition application.

For sound capturing it is recommended that the microphone should be as close to the speaker as possible. This will increase signal-to-noise ratio as the speech will be louder compared to the background noise. This also reduces the effects of the reverberation to the recognition. [29]

One benefit compared to the Aalto ASR is that Google's cloud speech recognizer has VAD built-in the system. This improves the performance especially when speech has pauses. In this application it could be for example a student thinking about the answer for the question.

Unlike Aalto ASR this system returns partial results. This means that the processing of the recognition results can be started before the speech ends. This is a very useful feature as the results include also confidence and stability parameters. The confidence parameter is value between 0–1 and it tells how likely it is that the result is correct. The stability parameter gives value between 0–1 as well and it estimates the probability that the recognizer will not change the results.

The main concern with the Google's recognizer is, its performance with children's speech. Since the system is closed, it is not possible to implement the speaker adaptation or anything else that helps if the performance is not good enough. Thus, it is really important to test the recognizer really well with children, to avoid the situation where the whole ASR part needs to be implemented again.

5.3 Text to Speech

In the development of the prototype the text-to-speech synthesis system was not evaluated in detail. The synthesizer was chosen as it seemed to satisfy the requirements to be understandable and response time to be short. The evaluation was done by using the prototype, without any scientific methodology. The system used in the prototype is the TTS implemented for Google translate service.

The specific evaluation of the system is left outside of the thesis as the quality of the sythesizer is highly subjective and thus the evaluation would need rather complicated listening experiments, that are out of the scope of this thesis. The response time was not measured but it is small compared to the time used for the speech recognition so its contribution to the total response time was relatively small.

6 Implementation

This section describes how the prototype done within this thesis is built. This section will cover each part of the system other than those that were presented previously in this thesis.

In this section first each part of the hardware and software used are presented and after that the additional services are introduced. Finally the building process and programming and other practicalities are discussed.

6.1 The Prototype

The prototype built for this thesis is a simplified examination system. The user can choose different categories of questions and then the application starts asking the previously defined questions. The prototype is ran with Raspberry Pi 3B (RPi) with external sound card to which the microphone and loudspeakers or headphones are connected. This system will be part of the final prototype which includes in addition the body and touchscreen and graphical user interface (GUI) for it.

The GUI is described in this section shortly to help the reader to understand how the system works. However, the implementation of the GUI is left outside of this thesis as it is not relevant to the topic.

6.1.1 User Interface

The user interface (UI) is composition of two different UI types: spoken and graphical UI. The basic idea is that most of the actions are done just by speaking as naturally as possible, but some actions such as the menu is done with GUI. GUI is used because when the user have to for example choose something from the list it is more practical and faster to show the options visually than go through them by speaking each choice.

The graphical interface as well as the program is divided into two separate modes. There are the menu from which the user can login or choose the different actions that are programmed into the prototype. The screenshot from the menu view is shown in Figure 12.

In the menus there are two different kind of buttons. The buttons with small text in the bottom are used for the system related actions like logging in ("kirjaudu"), presentation of the prototype ("esittely") and closing the program ("sulje"). The presentation is mode where the user can ask predefined questions about the system from the robot. Those questions include for example "Who are you?" and "What can I do with you?" and the system will present itself with the answers. In the middle there are the other buttons that are used to start different types of examination games. In the prototype there is "Demonstration" ("demo"), "Addition and Subtraction" ("plus"), "Multiplication and Division" ("kerto") and "Clock" ("kello") games.

The first of the exams, "Demonstration", is a collection of all the different type of questions that can be done with prototype to demonstrate the system. "Addition and Subtraction" and "Multiplication and Division" are math games where in the

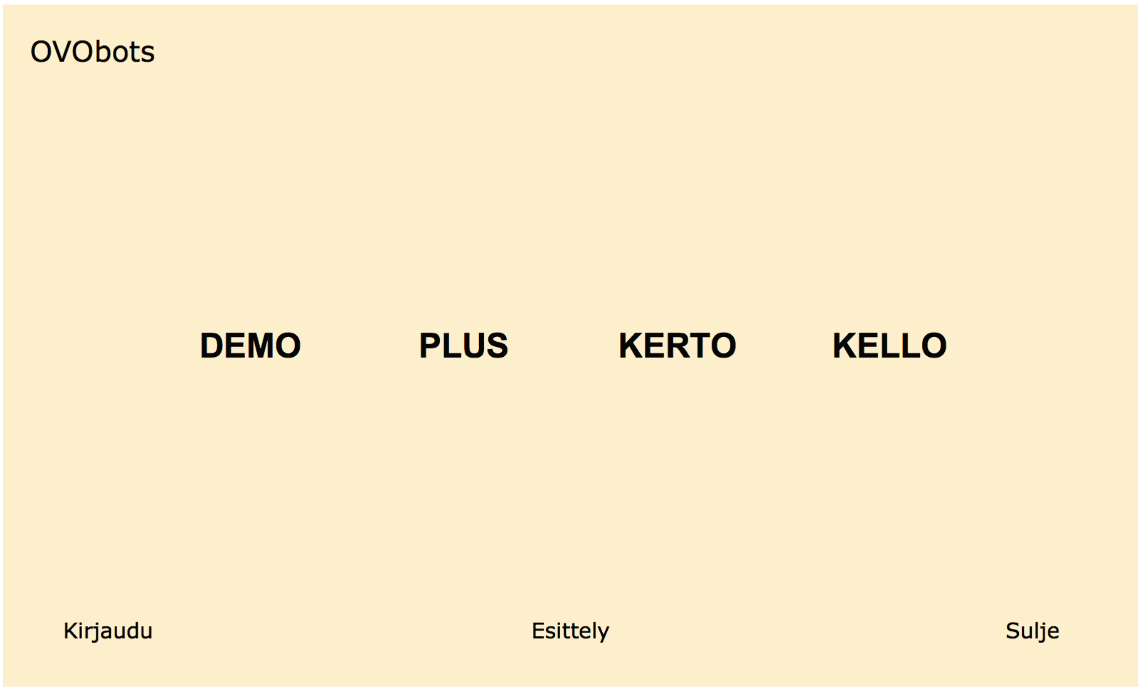


Figure 12: The menu view of the graphical user interface.

first only additions and subtractions are practiced and in the latter the results of multiplications and divisions are asked. The last game – clock – is about practicing how to read an analog clock. Thus, it shows pictures of different kind of clocks and asks the user to say what the time is.

The other mode is the examination view, in which the questions asked are shown on the screen and sometimes there are also pictures which are attached to the questions. The pictures of the examination with different types of questions view is presented in Figure 13. The color of the logo in the upper-left corner is used to inform the user when the system is recognizing. When the logo is black the recognition is not used and in most of the cases that is only when the system is playing some sounds. When the logo is green the system is recognizing.

6.2 Hardware

The hardware of the spoken dialog system is composed of two major parts. The first is the computer unit Raspberry Pi which is used to run the software. The other part is the sound hardware, which includes sound card, microphone, loudspeakers and amplifier. All the hardware choices are introduced in the following Sections 6.2.1 and 6.2.2.

6.2.1 Raspberry Pi

Raspberry Pi 3B is a small sized computer built on the single circuit board developed by a charity knowns as Raspberry Pi Foundation. It is developed to support teaching

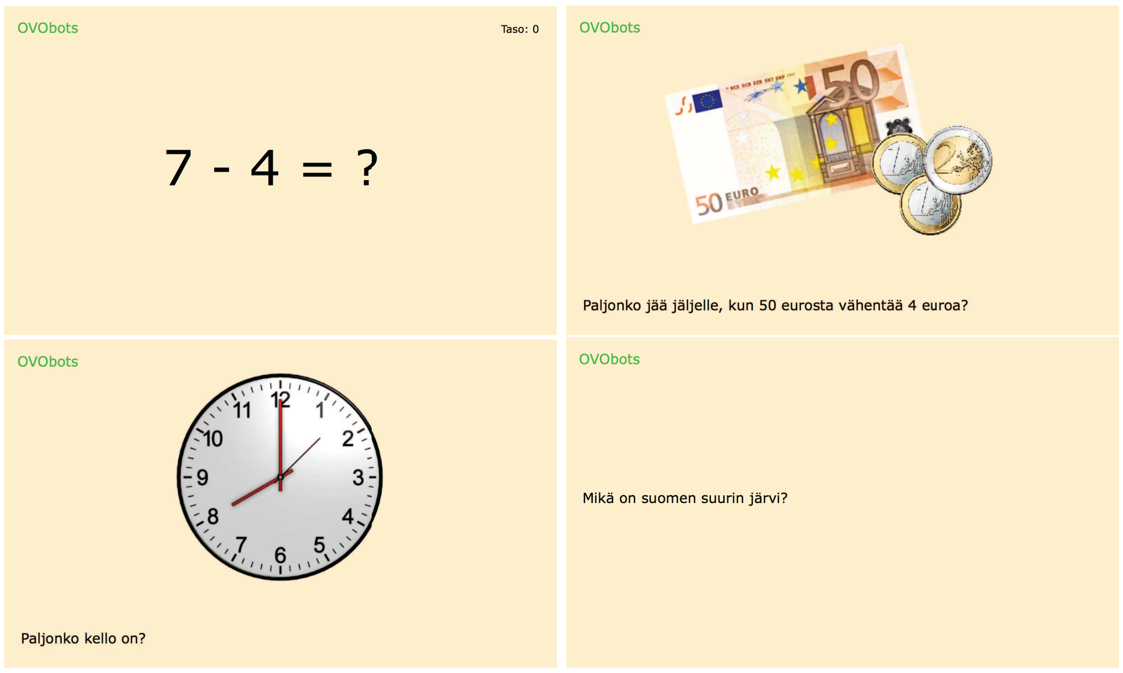

OVObots
Taso: 0
7 - 4 = ?
OVObots
Paljonko jää jäljelle, kun 50 eurosta vähentää 4 euroa?
OVObots
Paljonko kello on?
OVObots
Mikä on suomen suurin järvi?

Figure 13: The four different examination views of the graphical user interface. The color of the OVObots logo shows to the user when the system is recognizing

of computer science and it is very popular choice for hobbyists and educational projects related to computer science. Due to it's size it is easy to mount inside of the body of the robot, which has only a small space for a computing unit.

The computing power of the processor, memory, and storage are very limited with the RPi. Thus, Raspberry is used only for the simple tasks and all the demanding operations are completed with cloud services and the system is assumed to be connected to the internet all the time. However, when all the heavy tasks are done in the cloud, RPi is a really good choice to run user interfaces and other simple tasks.

In this project the GUI, natural language understanding, reply generation, audio capturing and playing and all the file management are done natively on Raspberry Pi. Only speech recognition and speech synthesis is done elsewhere.

6.2.2 Sound Hardware

Raspberry Pi does not include any sound hardware that is needed for the spoken dialog system. Thus, the parts have to be attached to the RPi separately via USB ports and general purpose input/output (GPIO) pins.

A microphone and a sound card should be evaluated by how much the different choices affect to the word error rate of the speech recognition. That is the most important measurement as the sound quality itself is not important if the recognition is as precise as with the better quality audio. However, this evaluation is left outside of the thesis.

The sound card and the amplifier used for audio output in the project are integrated into the single circuit board and the speech is captured with the USB microphone.

This choice is done, because it was easier to find good quality, affordable and readily obtainable sound cards for RPi that do not have sound input. The sound card used is Amp+ produced by HiFiBerry, which includes digital to analog converter and 25 W stereo audio amplifier. The loudspeakers were not yet specifically chosen at the time of the thesis.

6.3 Google Compute Engine Server

Google Compute Engine is a service that allows the user to build a virtual machine in the cloud. In this project this is used as a Aalto ASR server computer. The engine runs the server that was described previously in Section 5.1.4.

The configuration used for the project includes one virtual central processing unit (CPU) and 3.75 GB of random-access memory (RAM). The virtual CPU is on hyper-thread of the 2.3 GHz Intel Xeon E5 v3 Haswell hardware CPU. As the Aalto ASR is not implemented to use multi-threading only a one thread is enough for the server that is used only by a one client. In the future if there are multiple products using same server the configuration should be different.

For the communication between Raspberry Pi and the server is listening a transmission control protocol (TCP) port 5554 for the recognition requests. The client sends the request for the recognition and when the request is received the server executes the backend software. Then the server recognizes the speech in the audio that the client streams and sends the results when the audio stream ends.

7 Experiments

This section presents the experiments, the test-setups, and the data that were used for the experiments. Also the methods used for the evaluation are presented. The overview of the speech recognition systems and models and other details related to those are covered already previously in Section 5.

In the experiments two ASR systems are compared in terms of performance and accuracy. The two systems are Aalto ASR and Google Cloud Speech API. The speakers in the experiments are children, which make the results interesting as the systems and the models are designed for adult speech.

The accuracy improvement for the adult model recognition with the adaptation to children data is experimented with Aalto ASR. The results of the Google's recognizer are used as a reference, how well commercially available LVCSR system recognizes the speech. A comparison is made to find out which system is suitable for the application. Improvement in accuracy, when the adult acoustic model is adapted with the children's speech, is investigated

The adaptation was applied two different ways. The first was conventional speaker adaptation, where each speaker has it's own adaptation parameters calculated. The other adaptation applied was an average children adaptation, where the one set of adaptation parameters was calculated using data from multiple speakers. Those same parameters are then used with every speaker in the test data. The techniques used for adaptation are VTLN, CMLLR and VTLN+CMLLR and those are presented in detail in Section 2.8.

7.1 Test Set Ups

The models, systems and adaptation methods used are discussed in Section 5. The speech data used in the experiments is not captured by using the actual robot hardware.

The other difference to the previously described setups is in AaltoASR. In the experiments it is ran locally at the test computer for recognition accuracy experiments. However, that does not affect to the accuracy results as the models and algorithms used are the same. The local test with AaltoASR are made with the scripts from [30].

The acoustic model is trained with all adult speech data from Speecon project. The language model is the 6-gram morph model, with a vocabulary size of 8429 morphs. In addition to the language model the 2-gram morph lookahead model is used as well. The language model is chosen to be unlimited vocabulary, which is not optimal for the task experimented, but it makes the comparison to Google LVCSR more fair.

Transcripts of the audio files generated by the recognition systems are evaluated by the Sclite tool. This tool is part of the NIST Scoring Toolkit (SCTK). Sclite calculates the error rates presented later in this section, by comparing the hypothesis transcript, got from the recognizer, to the reference transcriptions, which contains the correct sentences in written form.

7.2 Experiment Data

The data used in the experiments is the Finnish speech data of kids from the Speecon project. The database includes sets of toy commands, phone number, dates, sentences from children's books and other similar utterances. The data used in the experiments is captured with a close field microphone. The speech is recorded in a home environment or on other similar environment, where there is non-stationary noise. The data includes recordings from close wall and far from wall. [26]

The children ages are defined in two groups, which are 8 to 10-year-old and 11 to 15-year-old. However, all the children data is recorded before a voice change. The groups are present in the data so, that at least 30 % of the speakers are required to belong each of the groups. For the children speakers there were no gender restrictions as the gender-based speaker differences are not evident. [26]

The data is divided to one test data set and two adaptation data sets. Figure 14 shows how the data sets are divided, between utterances and speakers. The groups of 15 and 35 speakers are constructed randomly. In the following sections the details of each set are explained.

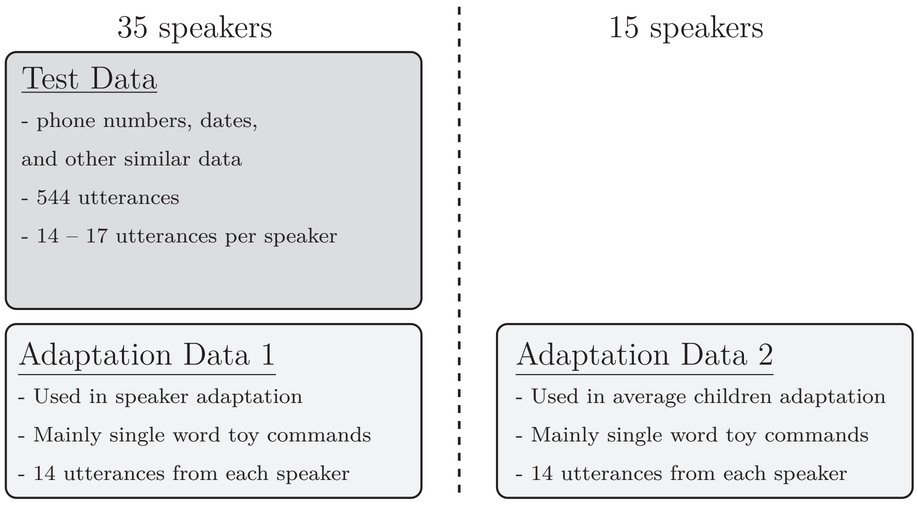

35 speakers
15 speakers
Test Data
- phone numbers, dates,
and other similar data
- 544 utterances
- 14 – 17 utterances per speaker
Adaptation Data 1
- Used in speaker adaptation
- Mainly single word toy commands
- 14 utterances from each speaker
Adaptation Data 2
- Used in average children adaptation
- Mainly single word toy commands
- 14 utterances from each speaker

Figure 14: The division of the data sets. The 50 speakers in the whole data are randomly divided in two groups of 35 and 15 speakers. The test data and adaptation data are not overlapping.

7.2.1 Test Data

The test data consists of 544 utterances and 2062 words in the speaker adaptation experiment. There is 14 – 17 utterances spoken by each speaker. In the test set 35 speakers are included and the data from the other 15 speakers are not used in the testing as the speakers are used in average child adaptation.

The data consists of utterances that are multiple words, but not complete sentences such as: phone numbers, single digits, dates, and expressions of time and money amounts. The data includes phrases with different types and amounts of background noise and sounds created by speaker that are not speech, like sighs and coughs. The data was chosen as it mimics the usage of the early stage prototypes better than random sentences.

7.2.2 Adaptation Data

The adaptation data is divided between the speaker adaptation and the average child adaptation data sets. The contents of the data sets are similar, but the speakers are different between the two sets. Both adaptation sets have 14 utterances from each speaker and the utterances are short and mainly single word toy commands.

Adaptation set 1, which is used in the speaker adaptation includes 35 speakers. The speakers in the se are the same speakers that are in the test sets. The total amount of utterances in the set is 490 utterances.

Adaptation set 2 consists of 15 speakers and 210 utterances. The average child adaptation is done by using the adaptation set 2. The speakers in the adaptation data 2 are not included in the test data sets.

7.3 Evaluation

This section presents the methods used for the evaluation of the ASR systems. The methods presented are used to measure the accuracy and the performance of the systems experimented.

7.3.1 Word Error Rate

The most used measure to evaluate speech recognition systems is word error rate (WER). As there is multiple different kind of mistakes, simply counting the incorrect words does not provide enough information about the recognition accuracy. In addition to incorrect transcription of the words, there could be added or missing words. WER takes these three recognition errors into account.

The word error rate is defined as:

$$\mathrm{WER} = \frac{W_S + W_D + W_I}{W_N} \cdot 100\%, \tag{22}$$

where W_S, W_D, and W_I are the counts of the words recognized incorrectly, the deletions and the insertions respectively. W_N is the amount of words in the correct utterance. In Figure 15 is example how the errors are are classified to the different error types. [9]

7.3.2 Letter Error Rate

In addition to word error rate letter error rate (LER) is used to evaluate the speech recognition results as well. It points out if the most of the words recognized incorrectly were completely wrong or was just the part of them wrong. If only part of the word is wrong that could be taken account in the keyword spotting, for example by searching,

```
Correct sentence:  This sentence   is recognized incorrectly

Transcription:     This sent sense is ---------- incorrect
                         S    I           D          S
```

Figure 15: An example how errors are classified when WER is calculated. S is substitution, I is insertion and D is deletion.

if the base words are correct and then the small mistakes made by recognizer could be omitted. This could improve the usability of the spoken language interface.

The letter error rate is calculated similarly to the WER presented in the previous section and the substitution, the insertion and the deletion are defined similarly. The only difference in the measure is that, when calculating LER each letter is processed as a separate word. Thus, even the same tools can be used to calculate it, if between each letter is applied whitespace. The sentence in Figure 15 is shown in Figure 16, but now the error classification is done for the LER.

```
Correct sentence:

T h i s _ s e n t e n c e _ i s _ r e c o g n i z e d _ i n c o r r e c t l y

Transcription:

T h i s _ s e n t _ s e n s e _ i s _ - - - - - - - - - - - i n c o r r e c t - -
                  I       S           D D D D D D D D D D D                   D D
```

Figure 16: An example how errors are classified with LER. S is substitution, I is insertion and D is deletion. _ represents white spaces in the text.

7.3.3 Real-time Factor

The real-time factor (RTF) is a measure to evaluate the speed of speech recognition systems. The measurement of the speed with RTF is made by comparing the time used for the recognition T_r to the duration of the speech T_s. The RTF is calculated with equation:

$$\mathrm{RTF} = \frac{T_r}{T_s}. \text{ [31]} \tag{23}$$

7.3.4 Statistical Significance

The statistical significance is used to determine if the results of two systems tested are significantly different, when the amount of the data is taken account. This means that the difference in the results is not only occurred by chance. In this thesis the significance of the results is determined by using Wilcoxon signed-rank test.

The Wilcoxon signed-rank test compares how many of the values are larger and smaller in the data sets given. In this thesis this means searching if the WER of different ASR configurations for each speaker is higher or lower and giving them ranks negative or positive based on the difference. If the amount of positive and negative ranks is too close each other, the difference is not considered to be statistically significant. [32]

In addition to that also p-values for each configuration pair are calculated. The smaller p-value indicates that the difference in results is more significant. The threshold of the significance is defined with α parameter, which is the upper limit to the p-value to be considered significant. In the experiments $\alpha = 0.05$. This means that there is only 5 % probability for each experiment that the results occurred by a chance.

8 Results

The section presents the improvements that were achieved using the different adaptation techniques and the different data for adaptation. The adaptation applied is supervised. As a reference in each table there are results of Google's commercial speech recognizer, acquired with same testing data. The language model weight used in the Aalto ASR was 35 as it gave the best accuracy in every experiment.

8.1 Speaker Adaptation

In the first experiment the speaker adaptation was applied to each speaker separately. Table 1 presents the experiment results for the speaker dependent adaptation method.

Table 1: Results for experimenting the speaker adaptation for each individual speaker separately.

Adaptation	WER (%)	LER (%)	RTF
None	8.1	3.1	0.7
VTLN	3.5	1.3	0.6
CMLLR	2.8	0.9	0.6
VTLN + CMLLR	2.4	0.7	0.6
Google	7.2	3.3	0.2

The transcripts of the Google's ASR were modified before the WER was calculated. This was done because the recognizer transcribes the digits as a numbers not as a words as in the reference transcripts. The recognition of time of the day is also different from the references as the sentence: "one o'clock in the evening" is usually transcribed as "13.00". For the calculation of WER that kind of transcripts were considered as the correct sentences, if the time of the day was the same as the meaning of a corresponding utterance.

From the results can be seen that Aalto ASR outperforms Google recognizer when the VTLN, CMLLR or the both techniques combined are applied in terms of the accuracy. The improvements in the accuracy when the adaptation is used were assumed to be significant before the experiments, as the base acoustic model is not presenting the speech of the children very well. Thus, applying any adaptation at all would improve the accuracy.

Even though, the Google's recognizer was not as accurate as Aalto ASR when the models were adapted, the RTF of the recognizer is much lower compared to Aalto ASR. This means that the response time of the recognizer is lower when using the Google Cloud Speech API. There was also a slight difference in the RTF measurements as Google recognition was used over the internet and Aalto ASR was used locally. Thus, it is likely that Aalto ASR will perform slightly worse when used in real use case compared to the measured values.

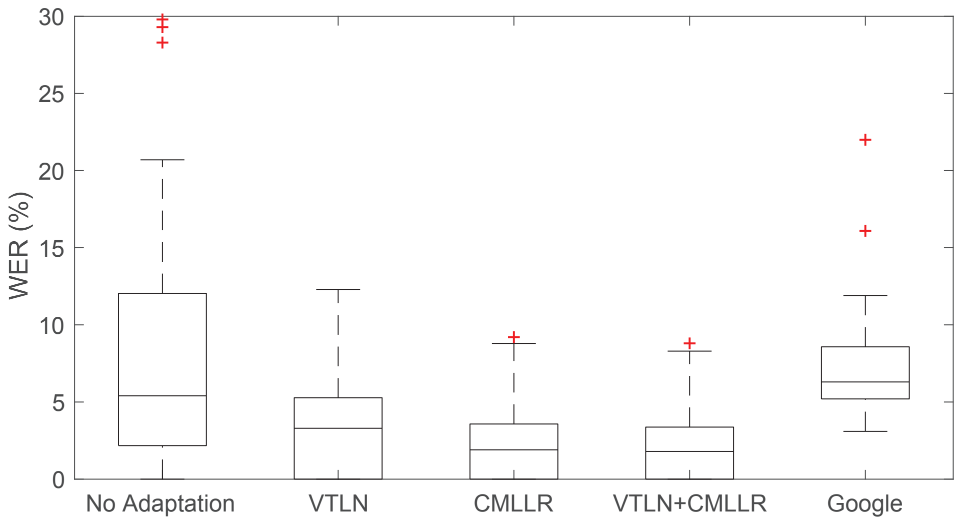


Figure 17: The division of the WER between different speakers, when speaker adaptation is applied.

In Figure 17 the word error rates of different speaker adaptation techniques are shown as a box plot of the different speakers. From the figure can be seen that applying the adaptation decreases the deviation between the speakers and also reduces the amount of the outliers in the data.

8.2 Average Child Adaptation

In the second adaptation experiment the error rates and real time factors were measured when the average child adaptation is used. This adaptation is done by using the same average adaptation parameters for all the speakers in the test data. The parameters are calculated using data from the speakers that are not included in the test data.

The results are shown Table 2 and the box plot between different speakers is shown in Figure 18. From the table can be seen that the SI adaptation outperforms the commercial recognizer in the accuracy, but the accuracy is not as good as with the speaker dependent adaptation.

Table 2: Result of the experiment done with the average child adaptation.

Adaptation	WER (%)	LER (%)	RTF
None	8.1	3.1	0.7
VTLN	3.5	1.3	0.6
CMLLR	4.1	1.3	0.6
VTLN + CMLLR	3.1	1.0	0.6
Google	7.2	3.3	0.2

The results show that unlike the speaker dependent adaptation the CMLLR performs worse than VTLN. CMLLR adapts to more specific features of the speaker than VTLN, which models the difference in the vocal tract length. It is possible that the effect vocal tract length varies much less than the other characteristics of each speaker in the test and the adaptation data. This is easy to verify by looking at the warping factors of each speaker as only two speakers had different warp factor 1.09 compared to the average 1.1. With the CMLLR this difference is not easy to see as the adaptation parameters are matrices with large amount of values.

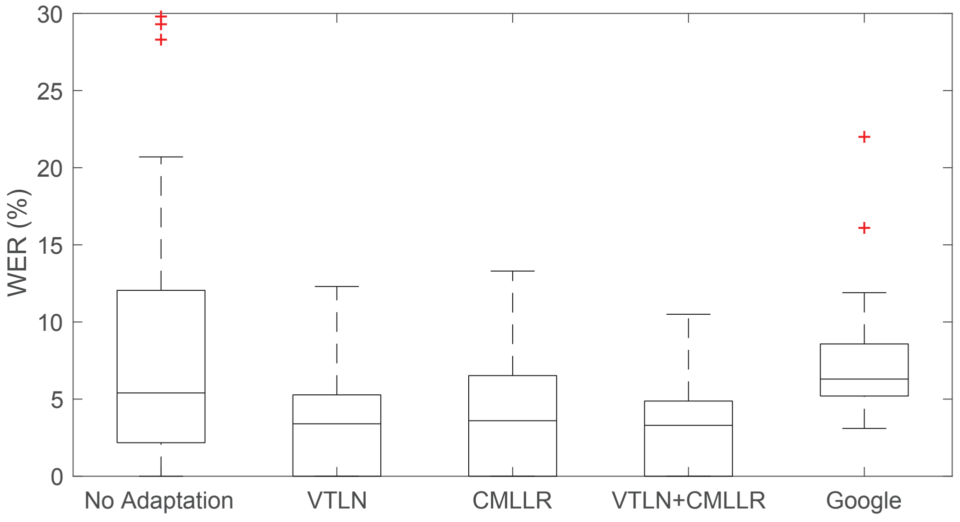


Figure 18: The division of the WER between different speakers, when average child adaptation is applied.

Based on the experiment results the automatic speech recognition system chosen to be used with the prototype was Google's. The choice was based on the speed of the recognition and the ease of use. The project is in the really early stage and the prototype had to be developed as quickly as possible so it was beneficial to use the system that has not be maintained internally.

8.3 Statistical Significance

To determine, if the differences in the accuracies of the systems were significant, the Wilcoxon signed-rank test was applied to the results of WER per speaker. The statistical significancy between each comparison pair is presented in Table 3. The table contains a p-value for each pair and the p-values that show the results to be insignificant are marked with red color.

From the statistical significancy table can be realized that all the differences when each adaptation methods compared to the not adapted model and Google recognizer appear to be statistically significant. Thus, it can be stated that accuracy of the adapted systems are significantly better compared to the Google and the non-adapted recognizer.

Table 3: Statistical significance matrix of the results calculated with Wilcoxon signed-rank test using WER per speaker. The p-values are shown in the table and the results that are not significantly different are marked with red color. $\alpha = 0.05$. There is + before the p-value if the configuration on the row is significantly better than the configuration on the column. When the method on the row is worse or the result is not significant there is no sign.

			Speaker Adaptation (SA)			Average Children Adaptation (ACA)			
		None	VTLN	CMLLR	VTLN+CMLLR	VTLN	CMLLR	VTLN+CMLLR	Google
	None		0.0014	0.0000	0.0000	0.0014	0.0001	0.0001	0.7315
SA	VTLN			0.0871	0.0049	1.0000	0.2948	0.3892	(+)0.0003
SA	CMLLR				0.4561	0.0674	(+)0.0032	0.0983	(+)0.0000
SA	VTLN+CMLLR					(+)0.0028	(+)0.0015	(+)0.0095	(+)0.0000
ACA	VTLN						0.3499	0.3289	(+)0.0004
ACA	CMLLR							0.0302	(+)0.0005
ACA	VTLN+CMLLR								(+)0.0000
	Google								

It is also worth to be noted that the speaker adaptation with VTLN+CMLLR gives significantly better WER than any other configuration tested except CMLLR speaker adaptation. However, according to the table the differences in WER between almost all of the other adaptation methods are not statistically significant.

9 Conclusion

In this thesis the development of the spoken dialog system for Finnish language was discussed. For the system automatic speech recognition, dialog modeling and speech synthesis are needed.

The main scope of the thesis is speech recognition and adaptation for children speakers. In the thesis the recognizer with the adult model adapted with children data was compared to the commercial recognizer. The speech synthesizer was implemented using an already existing commercial system, and the dialog modeling was done with the keyword spotting and with the prewritten utterances.

The experiments show that the Aalto ASR with adult acoustic model adapted with children data can be more accurate than the commercial recognizer. By using VTLN and CMLLR adaptations combined the word error rate can be decreased with speaker adaptation from 8.1 % to 2.4 %. The same technique for the average children adaptation decreased the word error rate to 3.1 %. In comparison Google's recognizer achieved 7.4 % error rate.

In practice the accuracy of the keyword spotting can be higher than the WER claims. This is because the most of the recognition errors were just the same word in different form like fifth ("viides") was recognized as five ("viisi"), which could be taken into account when keywords are searched. However, when evaluating and comparing the ASR systems it would be not practical to take that in account.

The statistical significance of the results was analyzed with Wilcoxon signed-rank test and the difference of WER of Google recognizer and both the speaker and the average child adaptation with VTLN+CMLLR was significant. The difference was also significant when the two adaptations was compared to the non-adapted Aalto ASR model.

Even though the accuracy of the Aalto ASR with adaptation was significantly better than the Google's recognizer, the latter was used in the prototype. This choice was done, because it was much quicker to process the recognition result and also it was much faster to get to the point where the prototype could be used with the Google ASR, which was important as the schedule was tight.

10 Future Work

With the current implementation the multiple answers can be detected by listing all the wrong answers in the question definition. However, doing this for every question takes quite a lot effort. There is two rather easy improvements that can be done to overcome this.

The first solution would be doing a task specific keyword spotting recognizer, that finds the keywords based on the phoneme probabilities. If there are more keywords in the recognition result than in the correct answer, the answer will not be accepted. However, in the current state when the questions are not in their final form, that would make adding questions more difficult. When the questions are developed further and are more static this would be useful improvement and also it would most likely increase the recognition accuracy.

The other way to overcome the same problem, could be using deeper natural language understanding to find out, which parts of the recognitions could be probably answers. For this word tagging can be used to see, if there is more words falling into the same category than in the correct answer.

To improve the accuracy of the recognition new acoustic and language models could be trained. However, to apply this speech data from children and text data that is close to the actual use case of the product should be gathered, which would need some working effort. After that it would be possible to train the acoustic model that is specifically made to be used with children and will most likely improve the accuracy.

The current language model is derived from the newspaper text, which is not similar to the actual usage. Using the proper data the improvement should be significant. However, with the current resources the improvement of the recognizer is not possible as it would take too much working hours to develop and maintain compared to the size of the company.

The quality of the speech synthesis could be increased by creating data bank of offline sythesized audio files. To save storage on the Raspberry Pi the files could be stored on the cloud server and new files could be generated whenever new phrases are added to the robot. By doing this significantly more time can be used to synthesize the speech, which improves the quality and the time used to get the response is relative to the length of the audio file. However, this would make the adding new questions and other phrases slower, which might be in some applications unacceptable.

References

[1] D. Giuliani and M. Gerosa, “Investigating recognition of children’s speech,” in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP’03). 2003 IEEE International Conference on*, vol. 2, pp. II–137, IEEE, 2003.

[2] A. Hagen, B. Pellom, and R. Cole, “Highly accurate children’s speech recognition for interactive reading tutors using subword units,” *Speech Communication*, vol. 49, no. 12, pp. 861–873, 2007.

[3] J. Gustafson and K. Sjölander, “Voice transformations for improving children’s speech recognition in a publicly available dialogue system.,” in *INTERSPEECH*, 2002.

[4] S. Strömbergsson, A. Hjalmarsson, J. Edlund, and D. House, “Timing responses to questions in dialogue.,” in *Interspeech*, pp. 2584–2588, 2013.

[5] J. Edlund, M. Heldner, and J. Hirschberg, “Pause and gap length in face-to-face interaction.,” in *Interspeech*, pp. 2779–2782, 2009.

[6] M. Heldner and J. Edlund, “Pauses, gaps and overlaps in conversations,” *Journal of Phonetics*, vol. 38, no. 4, pp. 555–568, 2010.

[7] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[8] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2nd ed., 2009.

[9] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1st ed., 2001.

[10] V. Siivola, T. Hirsimäki, M. Creutz, and M. Kurimo, “Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner,” in *Proc. Eurospeech*, vol. 3, pp. 2293–2296, 2003.

[11] S. Ortmanns, A. Eiden, H. Ney, and N. Coenen, “Look-ahead techniques for fast beam search,” in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 3, pp. 1783–1786, IEEE, 1997.

[12] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, *et al.*, “Automatic speech recognition and speech variability: A review,” *Speech communication*, vol. 49, no. 10, pp. 763–786, 2007.

[13] P. C. Woodland, “Speaker adaptation for continuous density hmms: A review,” in *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*, 2001.

[14] L. Lee and R. Rose, “A frequency warping approach to speaker normalization,” *IEEE Transactions on speech and audio processing*, vol. 6, no. 1, pp. 49–60, 1998.

[15] D. R. Sanand, R. Schlüter, and H. Ney, “Revisiting vtln using linear transformation on conventional mfcc,” in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[16] D. R. Sanand and S. Umesh, “Study of jacobian compensation using linear transformation of conventional mfcc for vtln,” in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[17] I. Szöke, P. Schwarz, P. Matejka, L. Burget, M. Karafiát, M. Fapso, and J. Cernockỳ, “Comparison of keyword spotting approaches for informal continuous speech.,” in *Interspeech*, pp. 633–636, Citeseer, 2005.

[18] H. Jiang, “Confidence measures for speech recognition: A survey,” *Speech communication*, vol. 45, no. 4, pp. 455–470, 2005.

[19] M. S. Seigel, *Confidence Estimation for Automatic Speech Recognition Hypotheses.* PhD thesis, Ph. D. thesis, University of Cambridge, 2013.

[20] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, *et al.*, “The htk book,” *Cambridge university engineering department*, vol. 3, p. 175, 2002.

[21] P. Taylor, *Text-to-Speech Synthesis.* Text-to-speech Synthesis, Cambridge University Press, 2009.

[22] M. Bisani and H. Ney, “Joint-sequence models for grapheme-to-phoneme conversion,” *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[23] T. Hirsimäki, M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkönen, “Unlimited vocabulary speech recognition with morph language models applied to finnish,” *Computer Speech & Language*, vol. 20, no. 4, pp. 515–541, 2006.

[24] T. Hirsimäki, J. Pylkkönen, and M. Kurimo, “Importance of high-order n-gram models in morph-based speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 724–732, 2009.

[25] S. Enarvi, *Finnish Language Speech Recognition for Dental Health Care.* Licentiate thesis, Aalto University School of Sience, Espoo, Finland, 2012.

[26] D. J. Iskra, B. Grosskopf, K. Marasek, H. van den Heuvel, F. Diehl, and A. Kiessling, “Speecon-speech databases for consumer devices: Database specification and validation.,” in *LREC*, 2002.

[27] P. C. Woodland and D. Povey, “Large scale discriminative training of hidden markov models for speech recognition,” *Computer Speech & Language*, vol. 16, no. 1, pp. 25–47, 2002.

[28] D. R. Sanand and M. Kurimo, “A study on combining vtln and sat to improve the performance of automatic speech recognition,” in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[29] Google Inc., “Google Cloud Speech API documentation.” https://cloud.google.com/speech/docs/, 2016. Accessed: 2016-07-21.

[30] P. Smit, J. Leinonen, K. Jokinen, M. Kurimo, *et al.*, “Automatic speech recognition for northern sámi with comparison to other uralic languages,” in *Proceedings of the Second International Workshop on Computational Linguistics for Uralic Languages*, The Research Group on Artificial Intelligence (RGAI), 2016.

[31] O. Cheng, W. Abdulla, and Z. Salcic, “Hardware–software codesign of automatic speech recognition system for embedded real-time applications,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 3, pp. 850–859, 2011.

[32] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric statistical methods*. John Wiley & Sons, 2013.