Developing a Conversation Assistant for the Hearing Impaired Using Automatic Speech Recognition

Juri Lukkarila

School of Electrical Engineering

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

Espoo 20.11.2017

Thesis supervisor:

Prof. Mikko Kurimo

Thesis advisor:

D.Sc. (Tech.) Kalle Palomäki

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Juri Lukkarila

Title: Developing a Conversation Assistant for the Hearing Impaired Using Automatic Speech Recognition

Date: 20.11.2017 Language: English Number of pages: 6+85

Department of Signal Processing and Acoustics

Professorship: Speech recognition

Supervisor: Prof. Mikko Kurimo

Advisor: D.Sc. (Tech.) Kalle Palomäki

Your abstract in English. Try to keep the abstract short; approximately 100 words should be enough. The abstract explains your research topic, the methods you have used, and the results you obtained.

Keywords: speech recognition

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Juri Lukkarila

Työn nimi: Keskusteluavustimen kehittäminen kuulovammaisia varten automaattista puheentunnistusta käyttäen

Päivämäärä: 20.11.2017 Kieli: Englanti Sivumäärä: 6+85

Signaalinkäsittelyn ja akustiikan laitos

Professuuri: Puheentunnistus

Työn valvoja: Prof. Mikko Kurimo

Työn ohjaaja: TkT Kalle Palomäki

Tiivistelmässä on lyhyt selvitys (noin 100 sanaa) kirjoituksen tärkeimmästä sisällöstä: mitä ja miten on tutkittu, sekä mitä tuloksia on saatu.

Avainsanat: puheentunnistus

Preface

This work was carried out during the spring and summer of 2017 at the Speech Recognition research group of the Department of Signal Processing and Acoustics at Aalto University School of Electrical Engineering. This thesis would not have been possible without the support of the Academy of Finland for the project *Conversation Assistant for the Hearing Impaired*.

First and foremost, I would like to thank supervisor Prof. Mikko Kurimo and my advisor D.Sc. Kalle Palomäki for giving me the opportunity to work on this interesting and multifaceted topic. I am grateful for the freedom and continued support given to me during this work. I would like to thank Symeon Delikaris-Manias and Juhani Paasonen from the Spatial Sound research group for assistance with the background noise recordings used in the user tests, and Ilkka Huhtakallio for providing the necessary audio equipment. My thanks to Olli Savisaari from the User Interfaces research group for consulting with user testing methods. From the Speech Recognition group, I would like to thank Seppo Enarvi, Reima Karhila, Katri Leino, Ulpu Remes, Aku Rouhe and Peter Smit for general help, tips and discussion.

I would like to thank Päivi Rainò from HUMAK and Tarja Kaikkonen from Kuuloliitto ry for helping to recruit deaf and hard of hearing persons for the user tests, and many thanks to the people who participated in the tests.

Finally, I would like to thank Siiri for all the support and motivation.

Helsinki, 1.11.2017

Juri Akseli Lukkarila

Contents

Symbols and abbreviations

Abbreviations

ASR	Automatic speech recognition
AVSR	Audio-visual speech recognition
DCT	Discrete cosine transformation
DNN	Deep neural network
DSP	Digital signal processing
GPGPU	General-purpose computing on graphics processing units
GUI	Graphical user interface
HCI	Human-computer interaction
HMM	Hidden Markov model
LER	Letter error rate
LVCSR	Large vocabulary continuous speech recognition
MFCC	Mel-frequency cepstral coefficient
OS	Operating system
PCM	Pulse-code modulation
RNN	Recurrent neural network
SNR	Signal-to-noise ratio
SPL	Sound pressure level
STT	Speech-to-text
UI	User interface
WER	Word error rate
WFST	Weighted finite-state transducer

Symbols

$\prod_{n}^{N} a_n$	Product from n to N, $a_n a_{n+1} \dots a_N$
$C(\boldsymbol{W})$	Frequency count for word sequence $\boldsymbol{W}$
f	Frequency [Hz]
m	Frequency [mel]
$\boldsymbol{O}$	Observation sequence (vector)
$P(\boldsymbol{W})$	Probability for the occurence of word sequence $\boldsymbol{W}$
$P(w_i\|h)$	Probability for the occurence of word w_i, given word history h
ppl	Perplexity measure
$\boldsymbol{W}$	Word sequence (vector)

1 Introduction

Understanding and participating in conversations has been reported as one of the most prominent problems that hearing impaired individuals face in their daily lives by a wide variety of studies and reports, ranging from medical research and engineering to sociology and beyond [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Experiences from previous projects at Aalto University's Speech Recognition group also point to the same conclusion. Conversations, and speech in general, form a major part of all human interaction, and the reduced or completely lost capability for conversations due to hearing impairment can have wide-ranging consequences not only on the hearing impaired individual directly, but also on the whole of society through social and economic factors [8, 5].

Losing this substantial part of social interaction can have a strong negative effect on a person quality of life and affect the opportunities available to them, for instance in education and employment [6, 9]. Therefore, it is ultimately a matter of equality and equal opportunities for the deaf and hard of hearing members of society. Furthermore, statistical studies on the prevalence of hearing impairment have reported that the burden of hearing loss has been constantly increasing around the world, and is presently higher than ever before [12, 3]. An estimated 500 million people around the world had disabling hearing loss in 2015, which translates to approximately seven percent of the world's population at that time [3]. New medical and technological solutions have been called for in order to effectively treat hearing impairment, as the number of afflicted people keeps growing worldwide [3].

Currently, no medical cure exists for the majority of hearing loss cases, in the sense that the normal biological operation of the ear is restored [1]. Sensorineural hearing loss is by far the most common type of hearing loss, and damage or abnormalities in the hair cells of the ear are a typical cause for it [1, 8]. Hair cells, which are responsible for translating the mechanical vibration of sound to electrochemical signals for the brain, do not regrow naturally, and once damaged, cannot be repaired with currently available medical methods [1]. Consequently, when a hair cell is damaged, the resulting loss in hearing is physiologically permanent, though modern treatments like the cochlear implant can partially restore hearing sensations [1]. Damage to hair cells is most commonly caused by exposure to excessive noise and through time by the aging process. Gene therapy and stem cell–based methods are being researched as a potential solution, and might someday enable the regeneration of hair cells [13], but at least for the foreseeable future no tangible cure is available. Instead, existing medical treatments rely on augmenting and amplifying the degree of hearing still present with personal electronic hearing devices, or in the case of severe hearing loss and deafness, through the surgically implanted cochlear implant that bypasses the outer and middle ear altogether [1]. In addition to the electrical devices focused on improving the level of an affected individuals hearing, spoken communication between persons is largely supported by translating sound to text and by using sign language, with both methods relying heavily on human translators

[1, 14].

1.1 Automatic Speech Recognition

Automatic speech recognition (ASR) is the science and technology of transcribing spoken language into written words automatically using computers [15, 16]. The history of speech recognition research goes as far back as the 1950s, when the first steps were taken at the renowned Bell Laboratories [17]. For a long time, working applications were comprised mostly of simple voice user interfaces and dictation systems with limited, application-specific vocabularies [15, 17]. The holy grail of speech recognition technology has arguably been real-time, speaker independent recognition of unlimited vocabularies, of which everyday conversational speech is a good example. In recent years, significant advances have been made especially in this area, referred to in ASR research as *large vocabulary continuous speech recognition* (LVCSR) [15, 18]. These advances have been enabled in large part thanks to the fast development and adoption of new machine learning techniques [15, 19]. In particular, *deep neural networks* (DNN) with multiple hidden layers (hence the word "deep") have proven to be of great value for ASR systems [15, 19]. Machine learning itself has become widespread and practical owing to the advances in computer hardware and computational resources available, especially through the popularization of the so called *general-purpose computing on graphics processing units* (GPGPU) technology for massively parallel computation, that has enabled efficient training of neural networks with relatively inexpensive and widely available computing hardware [15, 19].

As a consequence, automatic speech recognition has gained a great deal of popularity during the last few years, and is starting to be widely used in practice in the commercial landscape and everyday technical applications, from smartphones and tablets to personal computers [15]. Today, companies actively promoting their own speech recognition systems to average consumers include influential giants like Amazon, Apple, Google and Microsoft. Services based on speech recognition technology encompasses voice-controlled personal assistants, dictation, automatic captioning, voice-based search of audio and video content, and a wide variety of voice user interfaces [15, 20]. While there are still some specific, and in many cases, quite substantial challenges for automatic speech recognition, in general it has become accurate and reliable enough to be used in many practical applications requiring continuous recognition of large vocabularies [15, 18, 21]. This is demonstrated well by the speech recognition based services of the previously mentioned large technology companies, such as Apple's Siri voice assistant [20]. At the same time, mobile smart devices (phones, tablets and laptops) and wireless internet access have become ubiquitous in all developed countries of the world, offering a convenient platform for utilizing speech recognition technology in practically every conceivable place and situation [15, 21].

1.2 Conversation Assistant

Returning to the communication problems hearing impaired people encounter, having automatic real-time transcriptions of speech always available could potentially be extremely helpful in many of these problematic situations. Realizing this type of system offers a challenging, but well-defined practical application of modern ASR technology to a concrete problem: The goal of this work was to develop and test a conversational assistance application aimed for deaf and hard of hearing individuals, which translates speech into text in real time using automatic speech recognition. The intended purpose of this assistive application, henceforth referred to as the *Conversation Assistant*, is to help and support hearing impaired persons in conversations, and other situations where they are being spoken to, such as meetings, lectures and school classes. It is not intended to fully replace other personal assistive devices like hearing aids, but instead to supplement them.

The basic operation principle of the proposed Conversation Assistant is illustrated in figure 1, describing a typical conversational scenario where the Conversation Assistant could be used. Ideally, the Conversation Assistant could be used with any applicable smart device the user already owns, as long as the basic requirements are met. The Conversation Assistant approach is described in detail in section 3. Using automatic speech recognition technology to support deaf and hard of hearing individuals in some capacity has been proposed and tested already previously

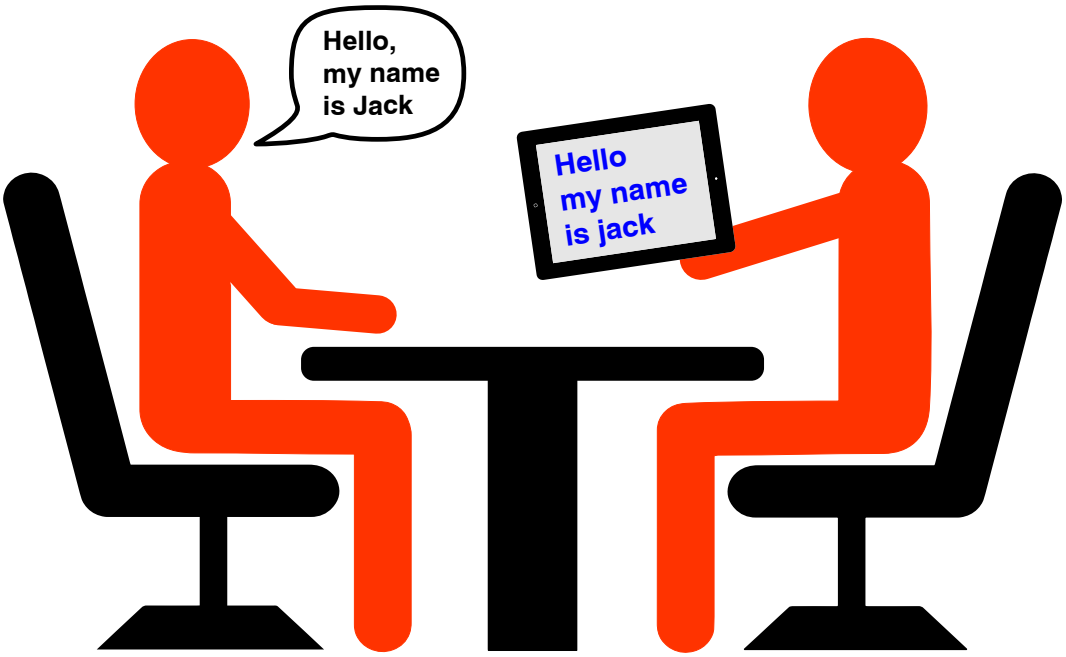


Figure 1: *The operation principle and an example use case for the Conversation Assistant. In the picture, the person on the left is speaking. The person on the right uses the Conversation Assistant, which converts that speech into text in real-time.*

[22, 23, 24]. However, many of these studies have focused only on some particular setting or situation, such as school classrooms, or have been otherwise limited in their scope. Also, automatic speech recognition technology has matured considerably during this decade, meaning that the findings of many earlier studies might not be accurate any longer. Furthermore, to the extend of our knowledge, this type of system has not been investigated or implemented before for the Finnish language.

1.3 Problems With Current Solutions

Some deaf and hard of hearing persons use sign language as an alternative for spoken communication. While this can work well for people who are fluent in it, the problem is that very few people know sign language, especially outside the deaf community. In fact, the share of sign language users among the hearing impaired has been constantly decreasing as a result of modern treatment technology [5, 14]. In particular, the cochlear implant has enabled a significant share of prelingually[1] deaf children to not require sign language for communication anymore [1, 2]. Different studies estimate that in the near future, approximately 60-80% of prelingually deaf children will use speech as their main means of communication, as opposed to sign language, thanks to the improved auditory perception provided by cochlear implants [14]. As a consequence, the need and incentive for the general public to learn sign language in order to communicate with hearing impaired individuals is diminishing even further. As the technological and medical solutions continue to advance, sign language is slowly becoming obsolete. Indeed, sign language is feared and predicted to become extinct in the coming decades [14].

While modern hearing aids and cochlear implants have become very sophisticated by utilizing digital technology and signal processing, there remains many challenges and limitations in their everyday use [25]. One of the major obstacles can be the cost: High-end, personalized devices and surgery are expensive, especially if not covered by health insurance or provided by a public healthcare system [3]. Adherence to hearing aid use and rehabilitation can be poor: It has been estimated that only 20-50% of people who would benefit from a hearing aid are actually using one [8]. Even with the extensive digital signal processing and noise reduction in current devices, speech intelligibility in situations with background noise remains one of the major challenges [7, 25, 26]. Cochlear implants in particular appear to be very susceptible to noise with a dramatic reduction in speech perception quality in noisy conditions [7, 27, 28, 29]. For public spaces and events, the effects of noise and the environment can be alleviated with a specially installed induction loop, commonly referred to as a *hearing loop* [30]: The desired sound source is fed electrically into a current-carrying wire loop, and the resulting electromagnetic field containing the baseband audio signal is picked up directly by a hearing aid or other device. The pickup coil in a hearing aid or implant is commonly referred to as a *telecoil* (or T-coil). Typical installation locations include airports, auditoriums, concert halls and public

[1] before language acquisition, including congenital cases (present at birth).

bureau buildings. FM systems are a similar alternative for induction loops, which use radio transmission instead of electromagnetic induction. Naturally, these solutions are also not without some technical and practical complications. Interference from metallic structures and other equipment can be an issue, leading to an uneven field strength and affecting the reception quality. One very concrete problem is that many places don't have them installed yet [7, 3].

Currently, human sign and written language interpreters have a large role in facilitating face-to-face communication between hearing and non-hearing persons, especially in more formal situations that can be scheduled in advance [14, 31, 32]. Written language interpreters translate speech to text simultaneously with a speaker by manually typing it into a computer. Written language interpretation happens only in one direction, whereas sign language interpretation can be bidirectional: first, the sign language interpreter translates speech into signs. Then the sign language user can respond with signs, which are then spoken aloud by the translator. Interpreters are used for example in classrooms, meetings, public events and private appointments. Requiring such an extra person for communication has many obvious disadvantages [32]. Firstly, there are the multitude of practical challenges: Interpreters are not available at all times and in all situations. Their number and availability can be quite limited especially outside urban population centers. Professional interpreters typically require a multi-year education and training, limiting their number and introducing costs. Privacy and self-determination can also be a concern, even though interpreters are customarily bound by confidentiality [31].

In conclusion, all the existing traditional solutions have their own problems. One of the fundamental issues with many of these are the costs associated with them, both for the individual and for society [3]. Our proposed, automatic speech recognition based solution has the potential to be a relatively inexpensive and highly cost-efficient alternative, enabling communication for the hearing impaired in most everyday situations quickly and conveniently. Ideally, the Conversation Assistant could also remove the need for human interpreters in many situations, or at least function as a workable alternative when human interpreters are not present or available.

1.4 Research Goals

A lot of research and progress has been made on improving automatic speech recognition technology [15, 17, 18, 19, 33, 34, 35]. While technical advancement and knowledge are valuable purely for their own sake, the practical application of this accumulated knowledge is arguably equally important. Correspondingly, the purpose and contribution of this work is to apply the latest developments in ASR into practice, in the hopes of helping with a real-world problem faced by millions of individuals around the world. The main focus of this thesis is on the practical implementation of a proof-of-concept prototype for the Conversation Assistant, as well as user testing the prototype with real users in order to properly validate it as a workable solution.

Consequently, designing and performing the necessary user tests for validating the concept and evaluating its usefulness for the end-users also formed a major part of the work. Overall, the contents and research goals of this thesis can be framed into four distinct segments:

1. Understanding the challenges deaf and hard of hearing individuals face in conversational situations.
2. Developing a proof-of-concept prototype for use in user testing.
3. Planning and carrying out user tests with real intended end-users.
4. Analysing the results:
   - Is the Conversation Assistant a viable concept, and feasible for practical use?
   - How it can, and should be, implemented?
   - What are the main factors for improving its usefulness?
   - Is there commercial potential for it?

1.5 Thesis Structure

This thesis is organized in the following structure: Section 2 gives an overview of the foundations of this work, providing background information on hearing impairment, automatic speech recognition, software development and user testing. It also includes a review of previous work relating to this topic and other proposed solutions for the same problem.

Section 3 describes the design and implementation of the Conversation Assistant prototype, including the software tools and libraries used in the programming. Likewise, the models used in our automatic speech recognition system and the data they were trained with are described briefly.

In section 4, the objectives, design quidelines and choices made for the user tests are presented, followed by a detailed description of the resulting test plan and its execution in practice. An objective comparison of the speech recognition models is done, where the models that were used in the user testing prototype are replaced with new, somewhat different models in the hopes of improving the speech recognition accuracy. The results from the user testing are presented and analyzed in section 5.

Section 6 concludes this thesis. It contains a summary of the work done and results obtained, together with a review whether the objectives set forth in the beginning were met. Finally, conclusions drawn from the results and avenues for future work are discussed.

2 Background

This section presents the theory and scientific context behind the work. Developing and testing the Conversation Assistant is a multidisciplinary task, requiring knowledge from a variety of fields from computer science and engineering to medicine and psychology. Firstly, since we are developing an assistive software-based solution particularly for deaf and hard of hearing individuals, comprehensive knowledge of hearing and hearing loss is required so that the problem being solved can be understood, and the factors affecting it taken into account. Likewise, it is equally important to possess a general overview of currently existing assistive solutions, in order to understand and assess the Conversation Assistants place and impact in the big picture. Information on the social impact of hearing impairment and the problems hearing impaired individuals face on a daily basis offer a clear-cut motivation and reason for the work presented. Additionally, it is relevant to know the demographics of hearing impairment in order to asses the scope of the problem and the scale needed for potential solutions, of which a very concrete example would be how many web servers could potentially be needed for a cloud-based speech recognition application for example in Finland. The number of hearing impaired individuals also directly affects the demand and commercial potential for the presented software solution. Functioning of the Conversation Assistant is based on automatic speech recognition, and therefore, it is necessary to understand how an ASR system works. The main goal of this work is to develop a software application that answers to the needs of the target user group as well as possible. Succeeding in this goal requires the discipline of software engineering, especially in the form of user-centered design and software engineering. Overall, these contents form the theoretical framework enabling the design, engineering and realization of the desired end-result.

The contents of this section are divided as follows: 2.1 contains a review of hearing impairment, including the physiological mechanisms and societal impact. In addition, existing treatments and assistive technology is covered briefly. Section 2.2 presents the theory and operation of automatic speech recognition systems, and section 2.3 describes the principles of software development, interface design, and user testing as related to this thesis. Previous work relating to this particular topic and other newly proposed technological solutions are reviewed in 2.4.

2.1 Hearing Impairment

Unlike many other disability groups, hearing impaired people are a highly heterogeneous group with different types and levels of hearing impairment [36]. There are some for whom sign language is their primary, or even possibly their only language, and conversely, there are many who do not know or use sign language at all [14]. Some are born deaf, while others can suffer from hearing loss later in life due to an illness, accident or through exposure to noise [1]. Age-related hearing loss is common for the elderly, even though many of them do not identify themselves as hearing impaired [30]. Typically, the term *hard of hearing* is used to refer to individuals

with some degree of hearing still present and communicate mostly through spoken language [37]. The term *deaf* most commonly refers to people with very little or no hearing ability at all, often using sign language for communication [37].

Hearing loss can be an invisible disability, and there remains some social stigma associated with it. Individuals with hearing loss often try to hide it, as it can be perceived to be associated with ageing or low intelligence [3, 30]. Access to modern assistive devices and treatment remains limited for many people, even in the developed countries [3]: The high cost of hearing aids and cochlear implants means that many people who could benefit from them cannot afford one.

2.1.1 Definition

Hearing impairment can be defined as having a reduced or deficient hearing ability, generally caused as a result of decreased hearing sensitivity in one or both ears [1, 4]. Neurological conditions affecting auditory processing in the brain have also been identified, though they are quite rare and hard to diagnose. These *auditory processing disorders* show as various difficulties in recognizing and interpreting sounds correctly, even though the ears are functioning physiologically normally [37]. The terms *hearing impairment* and *hearing loss* ar commonly used synonymously [1]. In this work, *hearing loss* is used to refer specifically to the sensory impairment of hearing, and *hearing impairment* to the overall condition and disability resulting from auditory dysfunction.

Hearing loss can be divided into two main categories based on the physiological cause [1, 37]: Conductive hearing loss is a result of abnormalities in the outer ear or in the ossicles in the middle ear, with the effect that sound is not properly transmitted to the inner ear. Sensorineural hearing loss results from a malfunction of the inner ear or the auditory nerve, meaning that the problem is in converting sound vibration into neural impulses and transmitting them to the auditory cortex of the brain. For example, damage to hair cells in the cochlea is a common form of sensorineural hearing loss [1]. The anatomy of the ear is presented in figure 2, describing the parts belonging to the outer, middle, and inner ear.

The overwhelming majority of all hearing impairments are of the sensorineural type, with age-related hearing loss (*presbycusis*) being the most common cause, followed by noise-induced hearing loss [8]. It is possible to have a combination of both conductive and sensorineural hearing defects as well. Another significant division for hearing loss and its treatment is the age of onset in relation to speech acquisition [37, 14]: Prelingual hearing loss is present before language acquisition, meaning it is either congenital (present at birth) or develops soon after. Postlingual hearing loss occurs after the development of speech and language. This distinction is important since deafness and hearing loss can significantly affect the language acquisition of children, a fundamental part of general cognitive development [3, 1, 14].

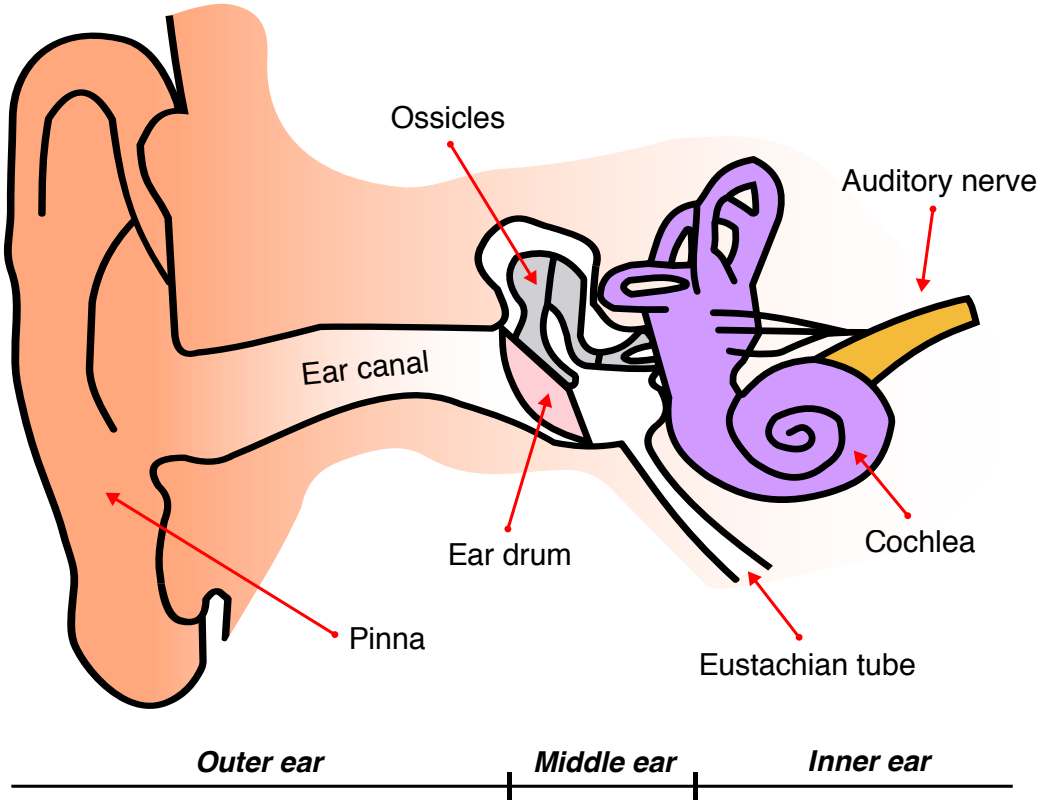


Figure 2: *Anatomy of the ear. The outer ear consists of the pinna and the ear canal, ending to the eardrum (tympanic membrane). The middle ear is a small air-filled cavity containing the ossicles: three tiny bones responsible for transmitting the vibration of the eardrum to the inner ear. The Eustachian tube is a narrow channel connecting the middle ear to the oral cavity, balancing the air pressure inside to the external air pressure. The inner ear houses the cochlea, a spiral-shaped liquid-filled tube containing the basilar membrane, along which the hair cells are positioned. Hair cells convert the vibration of the basilar membrane into neural impulses in the auditory nerve. [1, 38]*

Audiologically[2], hearing loss is categorized and its severity ranked according to the increase in the threshold of hearing, i.e., the sound pressure level required for the perceptual detection of sound [1]. It is measured in decibels and compared to the statistically defined and standardized nominal level of hearing. Human hearing is strongly frequency-dependent, and is most sensitive at frequencies from approximately one to five kilohertz [38]. Consequently, this frequency range is critical for speech perception and many other everyday tasks. Figure 3 presents the equal-loudness contours (curves) as they are defined in the ISO 226:2003 standard [39]. These contours describe the *sound pressure level* (SPL) required for a pure tone (single frequency sinusoidal waveform) to be judged equally loud depending on the

[2]Audiology is the scientific study of hearing, including the treatment of hearing defects.

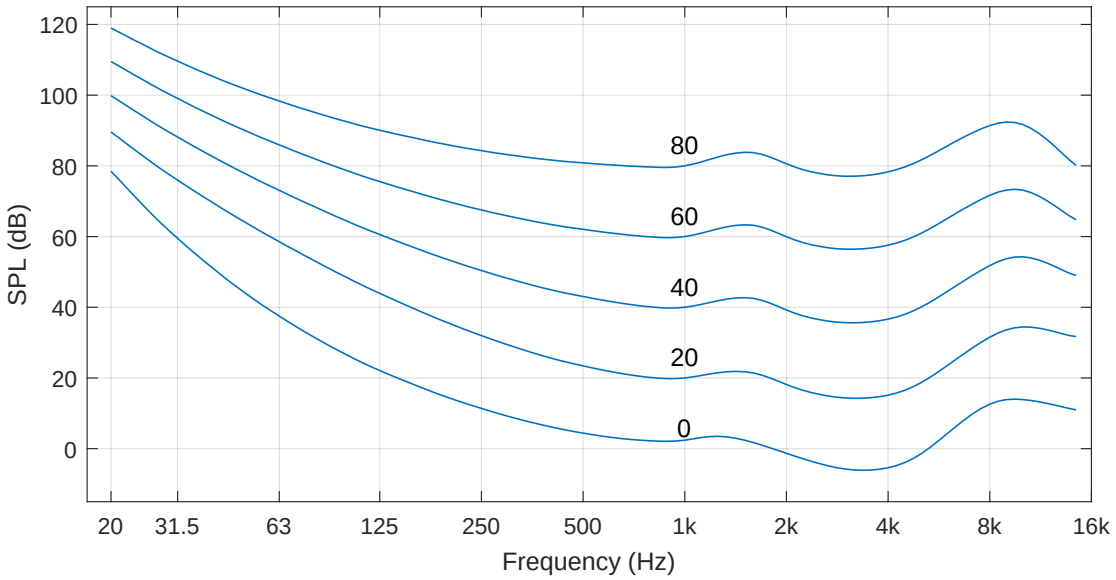


Figure 3: *Equal-loudness contours as defined in ISO 226:2003. The number associated with each curve is the nominal loudness level in phons, with zero phons corresponding to the nominal threshold of hearing. Sound pressure level is the pressure level compared to the reference value of* $20 \cdot 10^{-6}$ *Pascals. [39]*

frequency of the tone, illustrating the frequency-dependent sensitivity of hearing [38]. Notably, the equal-loudness curve for zero phons corresponds to the absolute threshold of hearing. Hearing loss can therefore be technically defined as upward changes to this curve, the frequency-dependent threshold of hearing, that exceed the normal small statistical variation between different persons.

The level of hearing loss is generally described as a single decibel value, referred to as the *pure tone average* (PTA) [1, 30]. It is calculated as the average of the hearing thresholds for pure tones at the frequencies of 0,5 kHz, 1 kHz, 2 kHz, and 4 kHz: For each frequency, the threshold of hearing is measured and compared to the reference value, resulting in a decibel value, which are then averaged together to form a single value estimate. Different severity categories have been set up with each category having a corresponding decibel range of the average hearing threshold increase. These decibel ranges vary slightly between countries and organizations [30]. Table 1 presents the common severity ranks, and the corresponding decibel ranges for the pure tone average used by the *World Health Organization* (WHO) and the *European Working Group on Genetics of Hearing Impairment* (EUWG) [30].

However, this does not mean that all individuals in the same category or even with the same pure tone average value have the same effective hearing impairment: Difficulties for instance with speech perception can vary widely depending on how much each frequency is affected and the mechanisms causing the hearing loss. People with mild or moderate hearing loss can typically understand speech

Table 1: *Categories for the grade of hearing loss and corresponding pure tone average ranges as defined by WHO and EUWG [30].*

	WHO	**EUWG**
Severity	Pure Tone Average (dB HL)	
Normal	0-25	0-19
Mild	26-40	20-39
Moderate	41-60	40-69
Severe	61-80	70-94
Profound	$\geqslant 81$	$\geqslant 95$

reasonably well in a quiet room with only one person talking. However, they can begin to have difficulties when more than one person is talking at the same time, or when there is background noise or notable reverberation present [2]. People with severe or profound hearing loss usually have difficulties even with understanding a single speaker in a quiet room, and severe problems when background noise is present. In addition to decreased audibility, hearing impairment typically causes additional perceptual difficulties not purely related to the reduced sensitivity [1]: The frequency resolution of hearing is often also affected, leading to the significant difficulties with speech perception in noisy conditions.

Normally, humans are very good at understanding speech in noise and can concentrate on a specific speaker or audio source among many other competing sources. This is typically referred to as the *cocktail party effect* [40]. In hearing loss, the auditory filters of the ear that divide the incoming sound into different frequency bands can become wider and more shallow due to damage to the hair cells [1]. This leads to increased auditory masking of adjacent frequencies, where one sound interferes with the detection of another sound. Also, the level of hearing loss typically varies between the ears, with one ear being better than the other. This affects the binaural processing done by the ear, which is important for sound localization, i.e., the detection of the direction and distance of sound sources [1, 38]. The end result is that the auditory system is unable to effectively isolate speech from noise, meaning that simply amplifying all sound does not help [1, p. 233–234]. Instead, the *signal-to-noise ratio* (SNR), the ratio between the level of the desired audio signal and the level of the background noise, needs to be improved considerably for better speech intelligibility in noisy conditions [7, 41].

2.1.2 Prevalence

In this case, prevalence is defined as the percentage of a population that is affected by hearing loss. A pure tone average greater than 25 dB HL in both ears is defined as disabling hearing loss by WHO criteria, as exceeding this level begins to clearly

impair communication in daily life [30]. Most population studies use this number for defining hearing impairment [30, 42]. From a public health perspective, hearing impairment can be considered a major health and economic problem: approximately 0,1 - 0,3% of newborn children have hearing impairment, while over 50 percent of the population over 75 years of age has a hearing loss requiring treatment [30].

In Finland, an estimated 11 percent of the workforce and around 15-18% of the whole adult population has some degree of hearing loss [8, 30]. The prevalence of hearing impairment among the elderly is high: A Finnish study with 4067 participants found that for the age groups 70, 75, 80, and 85 years, the prevalence for at least mild hearing loss varied between 37,7% - 54,1%, and between 21,1% - 38,9% for moderate or more severe hearing loss [30]. Among this group, hearing aids were used daily by 55,4% of the 249 persons who responded to a mailed interview.

In the USA, Lin et al. [42] estimated in 2011, based on data from 2001 to 2008, that 30.0 million or 12.7% of Americans aged 12 years or older had bilateral hearing loss, increasing to 48.1 million or 20.3% when individuals with unilateral hearing loss were included. Overall, they found hearing loss to increase with every age decade, with the prevalence of hearing loss expected to rise because of an aging population.

According to the Global Burden of Disease study of 2015 [3], approximately half a billion people have disabling hearing loss globally, which corresponds to 6,8% of the whole world's population. Wilson et al. [3] report that these numbers are substantially higher than estimates published before 2013, pointing to the growing importance of hearing loss as a disability, and the need for greater attention to global hearing health care.

2.1.3 Assistive Devices

The main technological solutions to hearing loss can be divided into three main categories: hearing aids, cochlear implants, and other assistive devices [1]. Hearing aids are used in mild to severe cases to augment a reduced hearing ability [1, 25]. A cochlear implant is required when hearing aids cannot help anymore, in cases of profound or total hearing loss [1, 25]. Other assistive devices include both supplementary techniques for hearing augmentation, and a wide variety of methods and gadgets based on visual perception and physical interaction. For additional hearing assistance, the previously mentioned audio induction loop and FM system are the most common and widely used, and many hearing aids and cochlear implants have a build-in receiver for these devices. Examples of the latter category include alarm systems using flashing lights instead of sound for applications such as smoke detectors and door bells. A typical example of an assistive device substituting sound with physical interaction is an alarm clock using vibration for awaking a deaf person [43]. Hearings aids and cochlear implants are being actively developed, and have improved remarkably during the past few decades, though neither device can match the performance of normal hearing [1, 25, 26]. More traditional support methods are

still being used as well: Lip reading is utilized by many hearing impaired individuals for understanding speech in conjunction with the devices.

Hearing aids are used to amplify and modify incoming sound, enhancing and augmenting the users own compromised hearing [25, 30, 41]. Modern devices are based on digital electronics and microprocessors, and come in many different types. The most common variations are the different *in-the-ear* and *behind-the-ear* models. Figure 4 presents various types of modern hearings aids from one particular manufacturer (Widex). In addition to the simple amplification provided by earlier analog models, modern digital hearing aids utilize real-time *digital signal processing* (DSP) for tasks such as speech enhancement, dynamics processing (compression), filtering, noise reduction, adaptive gain control and feedback cancellation [1, 25, 26]. Directional microphone systems are used for improving speech recognition in noise. Modern wireless transmission technology, such as Bluetooth, can be used to easily link hearing aids directly with sound sources like a telephone or television [30]. Modern hearing aids are personally fitted for each user, matching and compensating for the individual frequency response of the users hearing for best results [1].

A cochlear implant is a surgically implanted device that can provide auditory perception to people with severe or profound sensorineural hearing loss, even in the case of complete deafness [1, 2]. Cochlear implants bypass the ear altogether by directly stimulating the auditory nerve with electrical signals through electrodes inserted into the cochlea. All implants are comprised of two major elements: The external, detachable parts, and the internal, surgically implanted parts. The external parts

Figure 4: *Different types of modern hearing aids: behind-the-ear (back, middle), receiver-in-canal (back, left and rightmost), in-the-canal (front, left) and in-the-ear (front, right). Image source: Widex*

are usually removed for activities such as sleeping and bathing [2].

Similarly to hearing aids, the audio signal is acquired with an external microphone or picked up wirelessly. The signal is processed with a speech processor typically worn behind the ear. It is then transmitted wirelessly through the skin to the internal implant. The internal implant part consists of a receiver and the electrodes. The placement of these components is illustrated in figure 5, which presents the same anatomic view of the ear as in figure 2, but with an added cochlear implant:

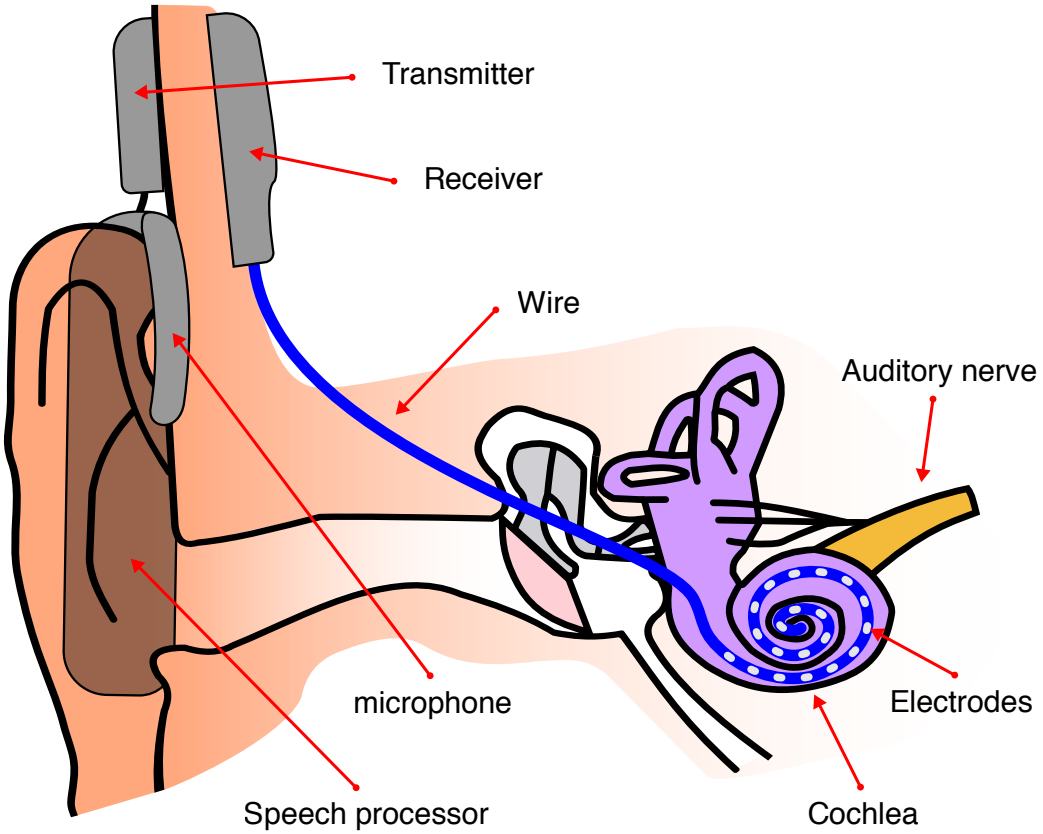


Figure 5: *An ear with a cochlear implant. A microphone and speech processor are typically worn behind ear. The speech processor connects to a wireless transmitter attached outside the head. An internal receiver inside the skin connects to the electrodes inserted into the cochlea. [1, 2]*

Most cochlear implants are multi-channel, meaning they have multiple electrodes. The input signal is likewise divided into multiple frequency bands with each electrode receiving its own signal [1, 27]. While they do not restore normal hearing sensation, cochlear implants enable speech communication for many recipients and significantly improve the spoken language acquisition for deaf children [5, 14]: Many

congenitally deaf cochlear implant recipients achieve a good speech perception ability and develop near-normal language skills. Post-lingually deafened implant recipients often regain the ability to understand and use spoken language at least to some degree. Cochlear implants have been shown to considerably improve the perceived quality of life for many recipients [10].

Cochlear implants are a relatively new treatment method: In Finland, the first few implants were installed in the mid 80s to adults. The first child was implanted in 1995 and the first congenitally deaf child in 1997 [14]. Today, the majority of prelingually deaf children are implanted already at the age of one, as the age of implantation is strongly associated with successful outcomes [2, 5, 14].

2.1.4 Social impact

Hearing loss can cause profound social problems and greatly affect an individuals physical and psychological well-being as a consequence of the difficulty with communication and social interaction. [3, 8]. The effects of hearing loss can lead to social isolation and stigmatization, depression, and problems with self esteem and work capacity [3, 8, 9]. Coping with hearing loss can be challenging, and consequently, psychological illnesses have been found to be more prevalent for the hearing impaired than for those in the general population [3]. On a personal level, the quality of life of an individual is ultimately impacted [10]. From a societal and public economy viewpoint, the consequences of hearing loss appear as productivity losses, employment issues, health care and social benefit expenditures, and reduced tax revenue [3].

Employment issues is one of the major obstacles hearing impaired people face in society [6]. In Finland, the unemployment rate of people with hearing impairment varied from 30 to 40 percent between the years 1995 and 2002 [6]. For young adults with hearing impairment, the unemployment rate was reported to be twice the rate of the normally hearing population in the same age group [6].

Employed hearing impaired people have been observed to have noticeable problems with coping at work and workplace well-being [3, 6]. The increased listening effort associated with hearing loss, particularly in noisy environments, can be tiring and cause stress [4, 8]. The negative effects of hearing loss are focused particularly to the aging workforce [8]. Consequently, there appears to be a clear statistical connection between hearing loss and early retirement [6].

2.2 Automatic Speech Recognition

In this work, the primary interest in automatic speech recognition is in using it in practice, instead of developing or improving upon some aspect of it. Therefore, this section focuses on providing a high-level, short overview of ASR systems. Likewise, detailed description of machine learning techniques and training methods fall outside this thesis. The focus in this thesis is on modern large vocabulary continuous speech recognition.

An automatic speech recognition system takes an audio signal containing speech as an input, and tries to produce the corresponding text representation [16, 20]. Figure 6 presents the high-level block diagram of a typical speech recognition system, though other configurations are also possible. In practice, the actual implementation can in many cases contain more complicated connections between the different components, and the models can be packed together for optimization reasons [15, 44]. The use of deep neural networks has also somewhat blurred the line between feature extraction and acoustic modeling [19, 44].

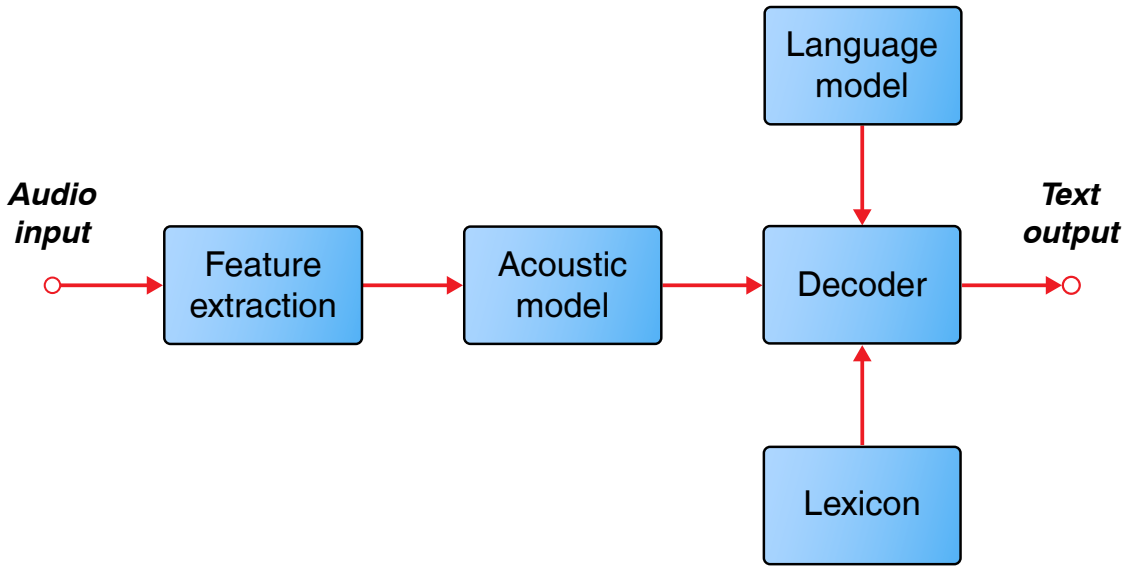


Figure 6: *Block diagram for the structure of a typical speech recognition system. The audio input can be a real-time signal from a microphone, or a existing recording or audio track. Depending on the application, the text result can be displayed immediately or saved into a text file.*

At the technical level, automatic speech recognition is a conversion process, where an acoustic data sequence (speech) is converted into a symbolic character sequence (text) [15]. In statistical terms, speech recognition can be categorized as a classification problem among the wider context of general pattern recognition tasks [16]. The speech recognition process can be formulated mathematically in the following way [16, 17, 44]: the goal of the system is to produce the word sequence

$\hat{\boldsymbol{W}} = w_1, w_2, w_3, \ldots, w_n$, for a given acoustic observation sequence $\boldsymbol{O}$. The most probable word sequence corresponding to $\boldsymbol{O}$ can be found by maximizing the posterior (i.e., conditional) probability $P(\boldsymbol{W}|\boldsymbol{O})$:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\arg\max}\, P(\boldsymbol{W}|\boldsymbol{O}) \tag{1}$$

However, $P(\boldsymbol{W}|\boldsymbol{O})$ is difficult to calculate directly [17], but it can be transformed into a form that is easier to model statistically. The Bayes' theorem states that [45, p. 10]:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{2}$$

With the Bayes' theorem, the probability $P(\boldsymbol{W}|\boldsymbol{O})$ can be transformed to the equivalent probability:

$$P(\boldsymbol{W}|\boldsymbol{O}) = \frac{P(\boldsymbol{O}|\boldsymbol{W})\,P(\boldsymbol{W})}{P(\boldsymbol{O})} \tag{3}$$

For finding the maximum, the denominator in equation (3) can be discarded as it stays constant, and equation (1) becomes:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\arg\max}\, P(\boldsymbol{O}|\boldsymbol{W})\,P(\boldsymbol{W}) \tag{4}$$

Here, $P(\boldsymbol{O}|\boldsymbol{W})$ is the probability of an acoustic observation sequence for a specific word sequence, corresponding to the acoustic model. $P(\boldsymbol{W})$ is the probability of a specific word or word sequence occurring, which is given by the language model. In practice, the probability product is often calculated as an addition in the logarithmic domain, with an additional weight α for controlling how much weight is given to the language model [17, p. 200]:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\arg\max}\{\log P(\boldsymbol{O}|\boldsymbol{W}) + \alpha \log P(\boldsymbol{W})\} \tag{5}$$

The following subsections present a brief review of each individual part of the speech recognition process presented in figure 6. In other words, how solving equation (4) is implemented in practice.

2.2.1 Feature extraction

The input to an ASR system is a digital audio signal, the digitized version of an acoustic sound wave and comprised of discrete samples [15, 16]. Figure 7 presents the waveform and the spectrogram for a short speech sample. The spectrogram is the magnitude spectrum of a signal as a function of time, describing how the frequency contents vary temporally. An audio signal contains acoustic information, such as the energy and frequency of its components. But in ASR, we are interested in the linguistic information. Therefore, the acoustic properties of an audio sample need to be somehow mapped to linguistic information. However, the problem with recognizing speech sounds is that every person has an unique voice, meaning that the same word or sentence spoken by different persons produce widely varying

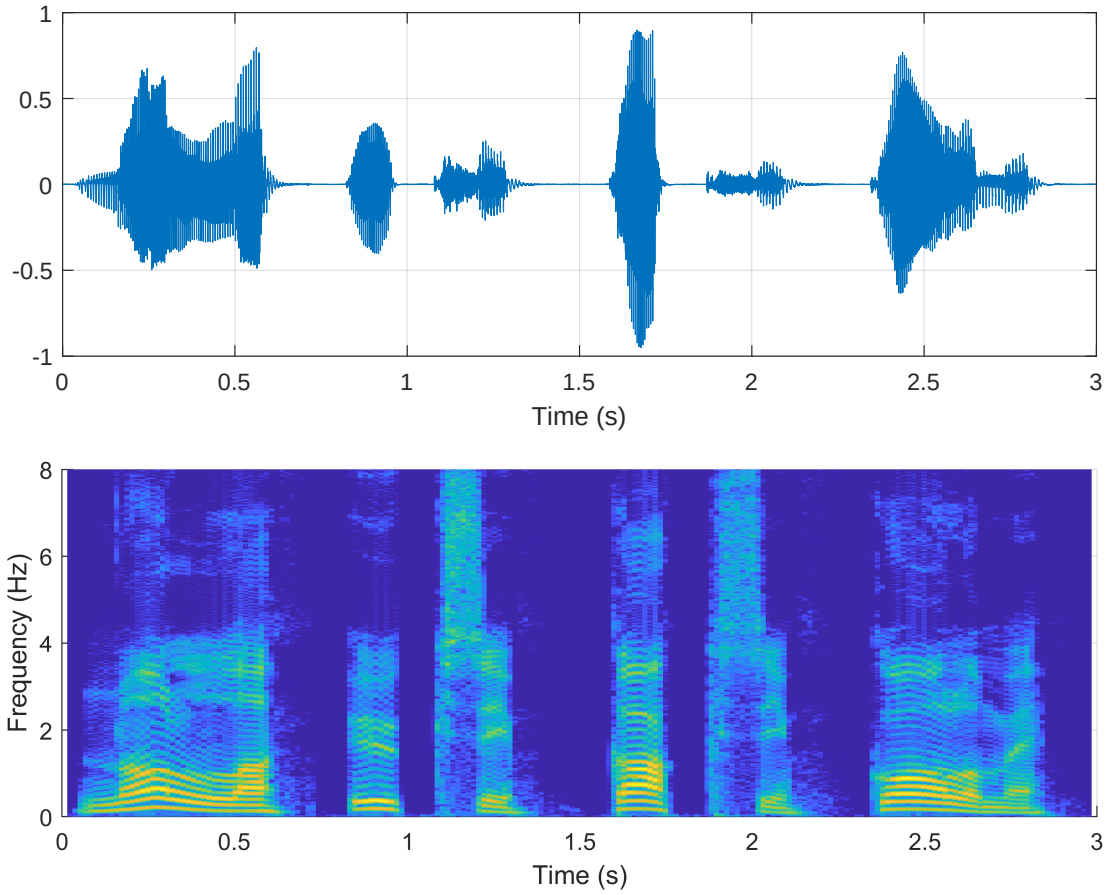


Figure 7: *The waveform and corresponding spectrogram of a speech signal containing the words "zero, one, two, three" pronounced in Finnish.*

acoustic information contents. These depend for example on the speakers gender, pronunciation, intonation, tone of voice, mood, and characteristic pitch and timbre. The voice of a one specific individual does not stay constant either, and can vary noticeably between different times and situations. Also, in addition to the speech signal of interest, all real-world signals contain some degree of background noise and other unwanted sounds from the recording environment [16]. Therefore, the acoustic properties of a speech sample cannot be used directly.

What is needed are some characteristic measures, or *features*, that can describe and discriminate between different speech sounds, and extract the linguistic information independent of the speakers personal voice. These features should contain the essential information and measurements needed for classifying speech. *Feature extraction* is then the process of calculating a sequence of these features, a *feature vector*, based on the input audio signal. Ideally, a feature vector should be compact, containing only the necessary information, be robust against noise, as well as fast to calculate so that it can be used in real-time [16]. Consequently, audio signals typically contain a lot of information that is not useful for this particular task. In

feature extraction, the input signal is therefore processed to remove unwanted and redundant information as well as noise, while emphasizing the important characteristics.

Common preprocessing steps include filtering the signal: The very low and high frequencies can generally be removed as speech is mostly focused between the frequencies from 100 to 4000 Hz. The input is often downsampled to a lower sampling rate, effectively removing unnecessary high frequencies as all frequencies above the Nyquist limit are lost. In practice, sampling rates as low as 8 kHz can be used, limiting the highest possible frequency to just 4 kHz, without a noticeably reduced recognition accuracy [16]. A pre-emphasis filter can be used to adjust the spectral tilt [44]. The signal is divided into partially overlapping frames, typically around 20 to 30 milliseconds in length, so that the frequency content of each frame can be expected to be fairly stationary [16, 17].

Mel-frequency cepstral coefficients (MFCC) has been a very popular choice for the feature representation type [15, 17, 44]. It is used because of its capacity for eliminating the speaker dependent characteristics of speech and for matching well to the logarithmic loudness and pitch perception of humans. The mel scale (short for *melody*) is a psychoacoustic frequency scale, where the change in pitch is judged to be perceptually equal in distance on the scale [38, p. 174]. Compared to the objective frequency scale in hertz, the mel scale puts more emphasis on low frequencies below one kilohertz, and conversely, compresses higher frequencies as the interval between perceptually equal pitch increments starts to increase exponentially with the frequency in hertz. The mel scale is used in order to better match the non-linear properties of human hearing, accentuating the frequency bands that are more important for human auditory perception. A common formula for converting a frequency in hertz to the mel scale is:

$$m = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right), \tag{6}$$

where f is the frequency in Hertz. The anchor point is set to 1000 Hz $\leftrightarrow$ 1000 mel [38, p. 174–175]. As we are interested in the frequency contents of the signal, each frame is windowed and converted to the frequency domain by calculating the short-time power spectrum [17]. A mel-scale filter bank with triangular band-pass filters spaced in perceptually equally long frequency bands is applied to each frame. A mel-scale filter bank is presented in figure 8. The combined energy of each frequency band is then calculated. Finally, the cepstral coefficients are obtained by first taking the logarithm of each energy band, mimicking the logarithmic loudness perception of hearing, and applying the discrete cosine transform, in effect compressing and decorrelating the energies [16, 17]. The cepstrum can be viewed as the spectrum of a spectrum, as the inverse (discrete) Fourier transform of a log spectrum transforms to the cepstral domain. In essence, the cepstrum operator deconvolves the speech signal into a linear combination of a source (excitation signal) and a filter, which can be then separated [16, p. 306–307]. In the *source-filter* model of speech production,

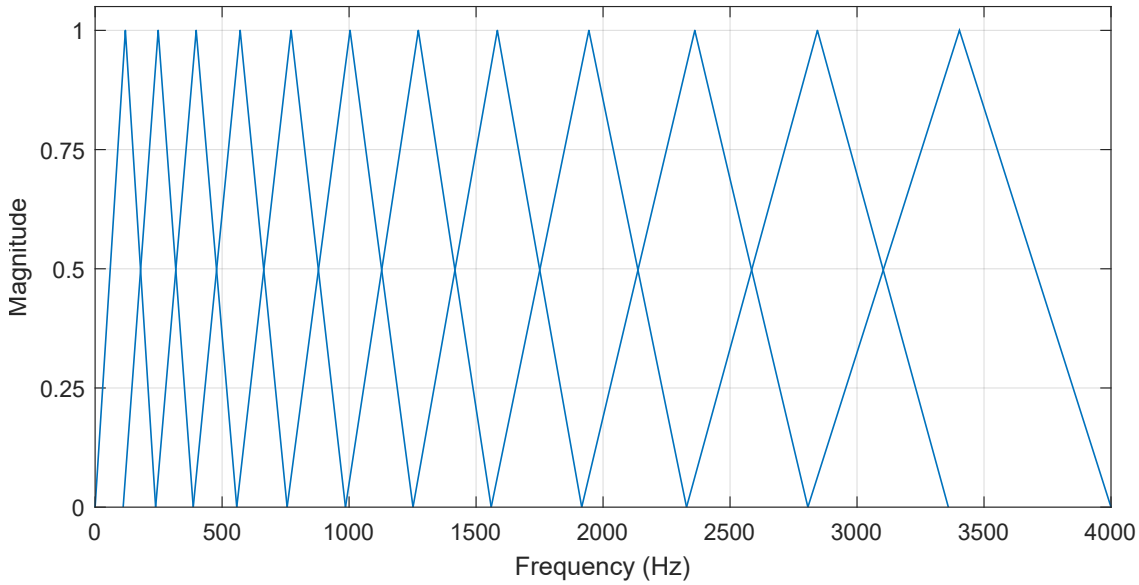


Figure 8: *Mel-scale filter bank with 12 filter bands from 0 to 4000 Hz. Note that in practice the number of filters is typically higher, a low number of filters was used here for a clear illustration.*

speech is formed by convolving a speaker-dependent excitation with a speaker-independent formant filter. We want to separate the speaker-independent filter part that corresponds to a specific phoneme [16, p. 288–290]. For an idealized case, this can be represented mathematically in the following way: Speech is modeled as the combination of a sound source (vocal chords) $e(n)$ and an linear acoustic filter (vocal tract) $h(n)$:

$$s(n) = e(n) * h(n) \tag{7}$$

The Fourier transform converts convolution in the time domain into multiplication in the frequency domain:

$$S(k) = F\{\, e(n) * h(n) \,\} = E(k) \cdot H(k) \tag{8}$$

Through the properties of logarithms, multiplication can be separated into addition:

$$\log S(k) = \log\big(E(k) \cdot H(k)\big) = \log E(k) + \log H(k) \tag{9}$$

A linear combination of the source and filter are then obtained with an inverse Fourier transform, which is a linear operation itself:

$$c(n) = F^{-1}\{\log E(k) + \log H(k)\} = F^{-1}\{\log E(k)\} + F^{-1}\{\log H(k)\} \tag{10}$$

In practice, the separation does not work perfectly, as the source-filter model is already a simplification in itself. However, it is often an accurate enough approximation for practical usage [16, p. 314]. The separation can be done with linear band-pass filtering, referred to as *liftering* in the cepstral domain. In practice, this can be achieved by applying a rectangular window to the cepstrum. Functionally,

this is the same as simply dropping the coefficients outside the filter band. Truncating the cepstral coefficients, i.e., taking only the n first coefficients, isolates the speaker-independent filter, meaning the spectral envelope displaying the formants of a speech signal. A traditional number of coefficients used is 12 [16, 17]. Conversely, the higher cepstral coefficients describe the primarily speaker-dependent excitation characteristics, corresponding to the spectral fine structure [17]. The effect of cepstral truncation on the spectrum is illustrated in figure 9.

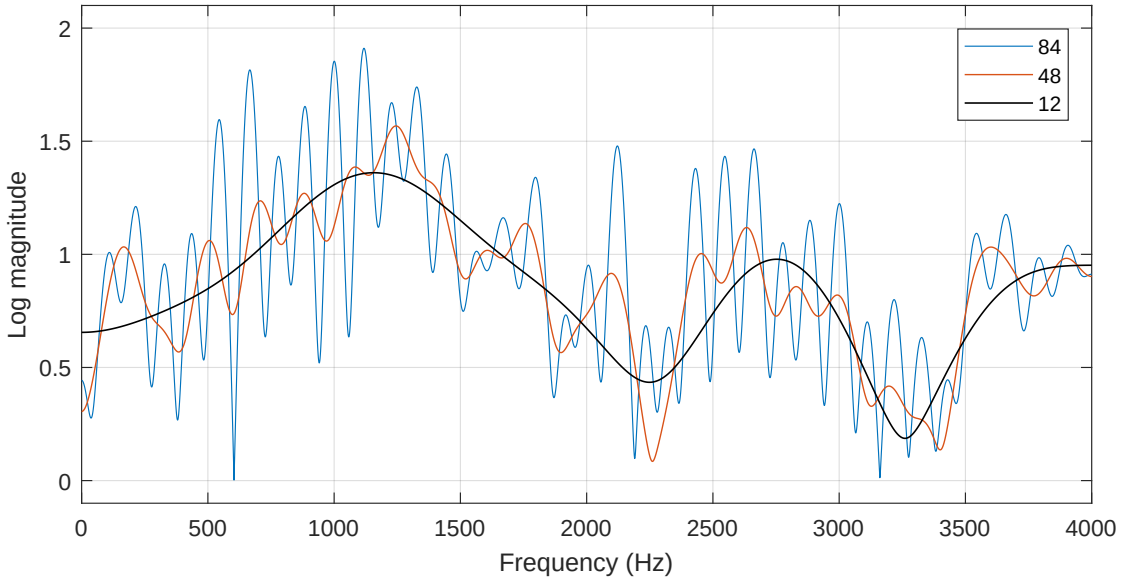


Figure 9: *The effect of cepstral truncation on the spectrum of a 30ms frame containing the Finnish phoneme /a/. The numbers describe how many cepstral coefficients are being used. As the number of coefficients is lowered, the underlying formant structure becomes clear.*

Additionally, the time varying nature of speech can be captured by taking the time derivatives of the obtained cepstral coefficients of consecutive frames [17, 44]. The first and second-order differences, referred to as the *delta* and *delta-delta* features respectively, are typically used for modeling the dynamic aspects of speech. The end result is a highly condensed set of features [16, 17]. The next step in the automatic speech recognition process is to map the acoustic speech features produced into linguistic classes, that can then be used to form the actual words. This is were the *acoustic model* comes in.

2.2.2 Acoustic model

The *acoustic model* provides a statistical model for classifying feature vectors into units of speech [44]. One of the practical choices for these classes are the *phonemes* of a particular language. Words or longer phrases could technically be used as well, however, this would lead to a very large number of classes as each word or phrase would require its own class. In written language, all words are formed with a very

limited set of alphabetic letters, or *graphemes*. In linguistics, a grapheme is the smallest unit of writing in any given language. For example, the English alphabet consists of 26 letters, and in Finnish 29 letters are used. Similarly, phonemes are the different units of sound that can be used to form all possible words in a particular language [16, p. 24–25]. For example, English is commonly divided into roughly 45 phonemes, varying slightly between different dialects and interpretations. This is very useful for acoustic modeling, since it means the acoustic model needs only to be trained to recognize this small set of phoneme classes, which can then be used to classify a practically unlimited number of different words. As the pronunciation of a phoneme depends quite heavily on the neighboring phonemes, it is common to actually use the triphone for the acoustic classes. A triphone is a sequence of three phonemes: one phoneme with the two nearest phonemes as context. Consequently, the acoustic model can be described as the collection of probabilities for how well a given feature vector corresponds to a specific phoneme sequence.

Hidden Markov Models (HMM) have been widely used for implementing the acoustic model [17, 45]. Speech is approximated well by a *Markov chain*, a stochastic model for randomly changing systems where the next state depends only on the current state [15, p. 23–26]. The chain consists of discrete states and state changes that are modeled with transition probabilities. In a hidden Markov model, the states cannot be observed directly (hence the term *hidden*). Instead, an output that is dependent on the states is known. In speech recognition, an utterance of a phoneme is considered a hidden state that emits an observable representation in the form of a feature vector. A three-state HMM is generally used to model one triphone in a left-to-right topology, meaning the states are constrained to be in chronological

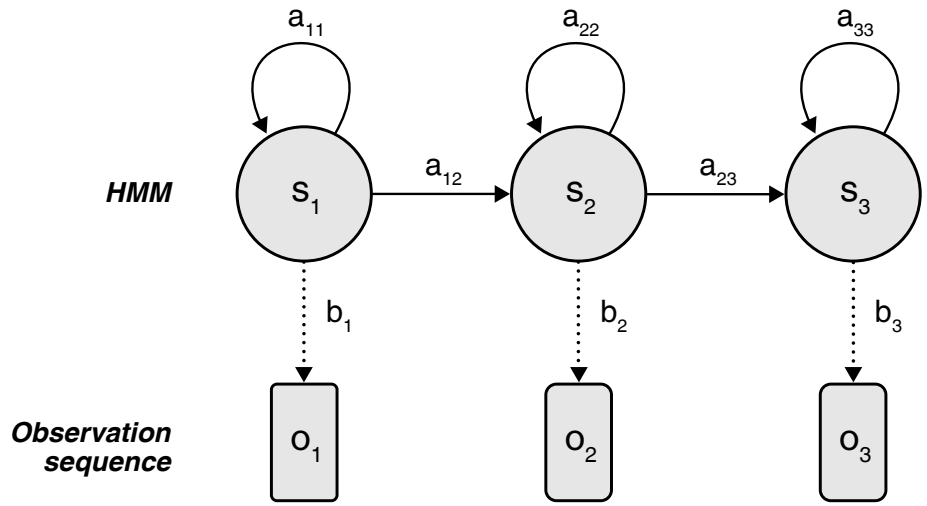


Figure 10: *A three-state, left-to-right HMM diagram. a_{ij} is the transition probability from state s_i to state s_j. Each state has an emission probability distribution b_i conditional over the observation sequence O.*

order [44]. Figure 10 present a three-state, left-to-right HMM. The probability that an observation is an emission of a specific state needs to be also modeled. For representing the probability distribution of observations, *Gaussian Mixture Models* (GMM) have been a popular choice [17, 19]. Gaussian mixture models are built by linearly combining two or more multivariate Gaussian distributions together. A simplified example of an GMM is presented in figure 11.

Like many other aspects in ASR, GMMs have been surpassed by deep neural networks in recent years [19]. A DNN is an artificial neural network that has two or more layers of hidden units between the inputs and outputs. DNNs are typically feed-forward, meaning data flows from the input layer to the output layer without looping back. They are trained with the *backpropagation* algorithm, where the gradient of a cost function measuring the error between the desired and produced output is propagated back to the network layers [15, p. 57–60]. Significant improvements to speech recognition accuracy have been achieved by replacing GMMs with DNNs [15, 19, 34]. The trend of using deep neural networks for acoustic modeling has somewhat affected the way feature extraction is implemented as well, with feature extraction incorporated as one aspect of the acoustic model: The extraction process can be started with a simple linear or mel-scale spectrum, and producing

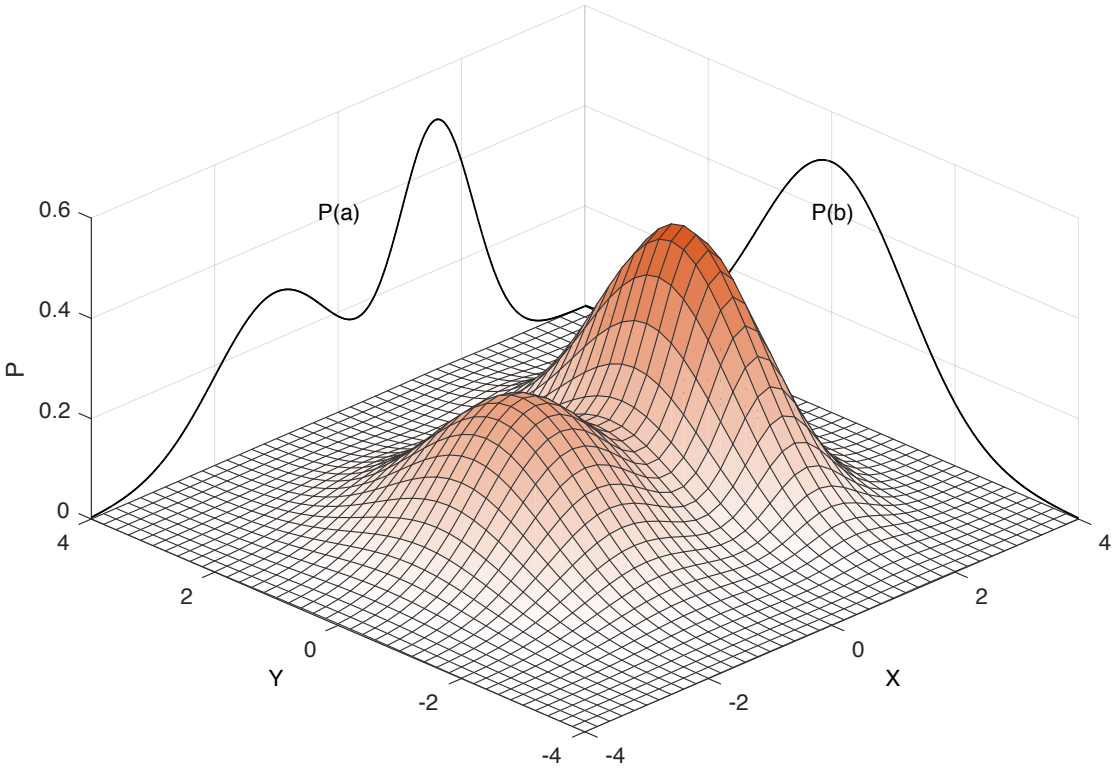


Figure 11: *A combination of multivariate Gaussian distributions. Note that the scaling is wrong for an actual GMM, as this is only a simple illustration. A much higher dimensionality is required as well in reality.*

the final feature representation type is relegated to the DNN, meaning the feature representation can be optimized as a part of training the acoustic model [44]. The ASR system used in this work is DNN based. A detailed description of the models and their training material is given in section 3.3.2.

2.2.3 Lexicon

The *lexicon* functions as a bridge between the acoustic model and the language model. It is used to describe how each word in the language model is pronounced by listing the phoneme sequence of every word. In other words, the lexicon can be defined as a dictionary for pronunciation in the context of an ASR system. Single phonemes (monophones) or triphones are normally used depending on the acoustic model. In Finnish, the pronunciation rules are very simple and each phoneme maps directly to one grapheme with only a few exceptions. Therefore, the lexicon can be generated automatically for the most part. For languages with highly irregular pronunciation rules, such as English, the lexicon is more complex and typically needs to be handcrafted by linguistic experts. Different pronunciations for the same word can also be accounted for in the lexicon. Table 2 presents an example of a triphone lexicon for the same four words in Finnish as in figure 7.

Table 2: *An example of a triphone pronunciation lexicon.*

Word	**Pronunciation**
nolla	_–n+o n–o+l o–l+l l–l+a l–a+_
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+_

2.2.4 Language model

The *language model* contains information on how words are used to form meaningful sentences in a particular language. Instead of grammatical rules about what combinations are theoretically possible, the language model tells which words typically go together and how often they are used. Specifically, it gives the likelihood for the occurrence of a specific word sequence, which is estimated from large collections of suitable written text called a *text corpus* [44]. Thus, the language model can be used to choose the most probable word sequence from multiple similar sounding candidates suggested by the acoustic model. Take for example the words *beer* and *bear*, which sound fairly similar to each other. While the sentence *"one bear please"* is grammatically correct and quite possible, it is typically much more likely that *"one beer please"* was uttered instead. The language model will likely assign a higher probability to the phrase *"one beer please"* (depending obviously on the data it was

trained with). This way, verbal context is provided for the recognition task, sorting the different hypotheses according to which are sensible and commonly used. The language model can also be used to efficiently reduce and rule-out unwanted and incorrect sentences, as word sequences with a probability of zero cannot be recognized.

Mathematically, the language model can be presented in the following way: Let $\boldsymbol{W} = w_1, w_2, w_3, \ldots, w_n$ be a word sequence. The objective of the language model is to estimate the probability $P(\boldsymbol{W}) = P(w_1, w_2, w_3, \ldots, w_n)$, the likelihood of that specific word sequence appearing in a particular language. Applying the chain rule of probability gives:

$$P(\boldsymbol{W}) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\ldots P(w_n|w_1, \ldots, w_{n-1}) \tag{11}$$

This means that the conditional probability $P(w_i|w_1, w_2, \ldots, w_{i-1})$ must be estimated for every word w_i, which is generally done using *maximum likelihood estimation* [46]. The estimate can be calculated as the ratio of the occurrences of the word sequences $w_1, w_2, \ldots, w_{i-1}, w_i$ and $w_1, w_2, \ldots, w_{i-1}$:

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) = \frac{C(w_1, w_2, \ldots, w_{i-1}, w_i)}{C(w_1, w_2, \ldots, w_{i-1})}, \tag{12}$$

where $C(\boldsymbol{W})$ is the frequency count of a word sequence $\boldsymbol{W}$ in a large text corpus. Due to data sparsity, this estimate has traditionally been too unreliable for long word sequences [17]. As an approximation, a word w_i can be assumed to depend only on a limited number of previous words. This is the *n-gram model*, where the conditional probability is truncated to depend only on the $n-1$ preceding words:

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) \approx P(w_i|w_{i-n+1}, \ldots, w_{i-2}, w_{i-1}), \tag{13}$$

with n ranging typically from two to four [17, p. 210]. The n-gram model has been the preferred way to construct language models for a long time [17, 44, 47]. In practice, the n-gram probabilities estimated with equation (12) need to be balanced, since n-grams that are present in the training material tend to get a too high probability and unseen sequences a too low probability. This process is called language model *smoothing*, and can be achieved by distributing some probability mass from seen n-gram combinations to all unseen n-grams [17].

In recent years, *recurrent neural networks* (RNN) have been shown to work well for language modeling [35]. They differ from the feed-forward neural networks typically used in acoustic modeling in that data can flow in any direction. Due to the recurrent connections enabling arbitrarily long context information in the network, RNN language models can capture the long-term dependencies ignored by the simple n-gram model [35, 48]. Neural network language models overcome the data sparsity problem by projecting words into continuous space, where the probabilities are then estimated [48]. While neural network language models offer improved performance for many applications, training them requires a great deal more computational resources compared to n-gram models, with the training time

typically measured in days or even weeks on high-end GPUs. Memory consumption of state-of-the-art neural networks can be an issue as well on currently available GPU hardware [35].

In agglutinative languages like Finnish, Estonian, Hungarian and Turkish, words are formed primarily by concatenating suffixes to root words, as well as using compounding (i.e., joining words together) and inflections (word bending) [46, 49, 50]. This means that there can be millions of regularly used word forms in these languages. Therefore, it is very hard to build a word-based vocabulary that would cover all the commonly used words [46]. Though it has recently become possible to build n-gram models covering millions of words, reducing the vocabulary size is important for efficient models [51]. For English, a 60 000 word lexicon can be sufficient for many tasks, whereas for Finnish, even a 500 000 word-based lexicon would not give the same level of performance [46]. The problem is the large number of *out-of-vocabulary* (OOV) words when dealing with limited vocabulary sizes.

Sub-word modeling has been successfully used to model agglutinative languages, reducing the size and complexity of the language model [35, 50, 47, 51]. In sub-word modeling, words are split into smaller units, which can be done based on grammatical rules or statistical techniques [46]. A data-driven statistical method called the *Morfessor* is commonly used for morphological segmentation. The Morfessor uses unsupervised machine learning methods to find morpheme-like statistical sub-word units called *morphs*, based purely on raw text data [46, 49]. A morph-based vocabulary was used in the Conversation Assistant prototype, as it is for Finnish speech recognition.

2.2.5 Decoding

In ASR systems, *decoding* means computing the result based on the input and the statistical models, i.e., solving equation (4), were the observation sequence $\boldsymbol{O}$ is the feature vector produced in feature extraction. The decoding process is fundamentally a search for the best matching word sequence over all possible word sequences, i.e., the *search space* defined by the models [33]. However, implementing the search in practice requires efficient algorithmic solutions: The size of the search space is often huge, and grows exponentially with the number of words in an utterance [33]. Therefore, an exhaustive search over all possibilities is generally not feasible. Heuristic methods are required to limit the search space in some way. For this purpose, *beam search* is commonly used [33, 45]. In beam search, only a limited number of most promising paths are kept while others are discarded in a process called *pruning*. The *Viterbi* dynamic programming algorithm is one such commonly used method [16, 15, 45].

ASR systems are divided into *online* and *offline* decoders [52]. In online decoding, the input is coming continuously in real-time and speech recognition is performed concurrently with the input. In other words, speech recognition is performed in

real-time and the results are typically displayed immediately. Conversely, in offline decoding the input is an existing (long) audio recording, which is then transcribed into text all at once. Offline decoding generally focuses on accuracy at the cost of speed and model size, though offline decoding can also be faster than real-time, meaning that the full transcription result is produced in less time than the length of the input file [53]. For example, a ten minute audio file could be transcribed in five minutes. On the other contrary, online decoding sets some restrictions for the properties of the ASR system, with the primary constraint being the requirement of approximately real-time recognition. Therefore, a compromise between speed and accuracy is typically required in online recognition. For example, the size of the vocabulary may have to be limited for fast enough performance [35, 21].

The use of *weighted finite-state transducers* (WFST) for representing the statistical models is popular in speech recognition, especially for LVCSR systems [45, 54, 51]. WFSTs can be understood as a finite automaton, consisting of a set of finite states and transitions between them. Each transition has an input label, output label and a weight for the transition. They are typically used to represent and store the acoustic model, language model and lexicon in one combined transducer, which offers many practical benefits. In the WFST–based speech recognition process, the acoustic model WFST transduces an acoustic state sequence into a phoneme sequence, the lexicon WFST transduces a phoneme sequence into a word sequence, and the language model WFST transduces a word sequence into a sentence. These transducers can be integrated into a single, large WFST that directly transduces an acoustic state sequence into a sentence. The power of weighted finite-state transducers for ASR comes from that they can optimize the search space and remove redundancies. The end result is a highly-optimized static structure for fast decoding, enabling real-time decoding with very large vocabularies of over one million words even on an average personal computer [45, p. 4–6]. The ASR system used in the Conversation Assistant prototype is a WFST-based online decoder, described in detail in section 3.3.1.

2.2.6 Evaluation metrics

Measuring the performance of an ASR system is a critical part of their development, so that different systems can be compared and new algorithms or implementations evaluated [16]. However, the answer to the question of how to measure speech recognition errors is not wholly unambiguous. It depends on what kind of occurrences in the transcription are defined as errors: For instance, are missing punctuation marks or capital letters counted as errors? In the recognized word sequence, there might be extra words added to the result or some words could be missing, in addition to the simple recognition errors were a word was transcribed incorrectly. Consequently, simply comparing two word strings one word at a time does not work: For example, if one word is missing or added as an extra, all the following words are interpreted as errors since they do not match the reference word in that position. Instead, the recognition result and the reference word string have to be aligned to each other [16].

Typically, the recognition errors in an ASR system are categorized as one of three main types of word errors [16, p. 420]:

- **Substitution**: a correct word was replaced with an incorrect word.
- **Deletion**: a correct word was omitted.
- **Insertion**: an extra word was added.

Based on these, the *word error rate* (WER) is defined as:

$$WER = \frac{S+D+I}{N} \cdot 100\%, \tag{14}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions and N is the total number of words in the reference text. Essentially, it is the *Levenshtein distance* for words, describing the total percentage of word errors. It should be noted that it is possible for the error rate to exceed 100%, as the number of additions is not limited. WER is the most widely used metric for evaluating and comparing the speech recognition accuracy of ASR systems [16, 44].

For agglutinative languages such as Finnish, the WER can be a too harsh measure: Many words are formed by adding affixes to a base word, resulting in a large amount of very similar sounding words with only a slight difference in their meaning. While technically incorrect, recognition errors of one or two characters in these words will most likely not hinder the understandability as much as the WER indicates. The *letter error rate* (LER) can be used as an alternative to the WER, giving a more accurate metric for agglutinative languages [44, p. 41–42]. In LER, the Levenshtein distance is simply calculated for individual letters instead of full words. Other measures can be used as well, particularly for measuring very specific types of errors or special vocabulary words, such as the error rate of foreign names or acronyms [48].

For evaluating language models, the *perplexity* measure is commonly used [48]. Perplexity measures how accurately a statistical model predicts a sample. In the case of the language model, this translates to how well the language model predicts a word sequence in an evaluation text sample. Perplexity is calculated by

$$ppl = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|h)}}, \tag{15}$$

where $P(w_i|h)$ is the probability of a word sequence given by the language model and N the number of words in the evaluation text. Minimizing perplexity corresponds to maximizing probability, meaning that a lower perplexity value is better [48].

2.2.7 Recognizing conversational speech

Accurate speech recognition of everyday conversational speech has been the focus of much research in recent years, and remains challenging especially in noisy environments [18, 44, 34, 20, 55, 33]. This particular area of ASR is the most relevant in the context of this work, as the Conversation Assistant is intended for use in everyday conversational situations. Conversational speech falls under the large vocabulary continuous speech recognition task, and is arguably the hardest area for online ASR. Conversational, or *colloquial* speech, is typically quite informal and can differ noticeably from the standard, formal form of the language [35]. In Finnish, colloquial pronunciations are also written differently than the standard word form, due to the phonetic orthography [35]. Consequently, colloquial speech causes more variations to language and increases the size of the vocabulary. For agglutinative languages, a vocabulary of many million words is required in order to cover all the spelling variations in conversational speech [35]. Overall, these factors have made it significantly harder to achieve a good level of speech recognition performance for conversational speech, compared to most other tasks.

Different dialects and pronunciations can pose a challenge to ASR performance, as the statistical models are only as good as the material they are trained with. Neural networks require a large amount of training data, and sourcing suitable training data covering the wide variety of distinct speaking styles can be difficult [56]. In order to recognize colloquial variations such as heavy accents, tens or hundreds of hours of material is typically needed for good performance. In addition, the recordings have to be accurately transcribed and aligned correctly. Even if recording are available, transcriptions are typically not, and producing them is generally expensive and time consuming [56].

Conversations are often held in acoustically challenging situations with background noise and other competing speakers. For large-scale everyday usage of ASR, noise-robustness has become an integral part for good real-world recognition performance [20]. More specialized ASR tasks can be performed in a quiet environment and with special, studio-grade microphone equipment, conforming to the limitations of the ASR system. For public usage, ASR systems have to cope with all possible environments and situations instead. Likewise, external microphone setups that could noticeably improve the SNR of the input signal are an unreasonable requirement for most everyday usage. Noise-robust speech recognition remains somewhat a challenge, though a large amount of research has been done to address this issue [44, 55, 18, 20, 57]. Fundamentally, the problem arises from distortions in the transmission path of the signal from the speaker to the microphone, with the effect that the observed feature sequence does not match the utterance of the speaker [44, 20]. These distortions can be from sources such as traffic noise, reverberation, background music, and overlapping speech. *Noise-robust* speech recognition methods attempt to reduce the mismatch between the noisy observations and the models [20, 44, 55]. One straightforward way to improve speech recognition in noisy conditions is to use

multi-condition training, where the acoustic model is trained with a combination of clean and noisy data. The feature extraction process can be improved further from the standard MFCC features to be more robust against noise. Additionally, *feature enhancement* methods have been presented that adaptively compensate for distortions in the features, such as using imputation methods for reconstructing missing data. Besides training noise robust models, the model parameters can adapted in real-time to the prevailing conditions [44].

2.3 Software Development

This section reviews briefly the software engineering principles and development methods relevant to this work. The focus in these areas is primarily on user-centered design and user testing. User-centered engineering [58].

User testing. Usability testing. Test planning and organization. User-centered agile method [59].

2.3.1 User-centered design

Design with the deaf [60].

Unified user interface design: designing universally accessible interactions [61].

Influences of new media communication on the deaf/hard of hearing as reflected in interaction design [62].

An experience in requirements prototyping with young deaf children [63].

Evaluating alternatives for better Deaf accessibility to selected web-based multimedia [64].

2.3.2 User testing

In the context of this thesis, usefulness is understood as to measure how well the system fulfills and performs its intended function, which in this case translates to how much and well does the Conversation Assistant help hearing impaired persons to follow and participate in conversational situations.

A practical guide to usability testing [65].

Handbook of usability testing [66].

On the contributions of different empirical data in usability testing [67].

Group usability testing: Evolution in usability techniques [68].

Experiences with usability testing [69].

Usability evaluation methods: Mind the gaps [70].

Usability testing of mobile applications: A comparison between laboratory and field testing [71].

2.4 Previous Work

In this section, previously presented novel approaches and solutions to the communication issues of the hearing impaired are reviewed. The main focus of this review is on research into the application of automatic speech recognition technology for helping in this problem. The idea of using speech recognition technology to support the deaf and hard of hearing is by no means completely new, as many different strategies for utilizing ASR in this context have been previously proposed and tested in practice. Using speech recognition as an assistive device for the deaf and hard of hearing has been proposed as early as 1996 [22]. ASR technology and recognition accuracy has improved considerably since the time many of these previous experiments and projects were done, which should be acknowledged when interpreting the results obtained and conclusions provided by them. Nevertheless, the previous research can still offer insights and historical perspective for the development of our Conversation Assistant.

One common focus in previous research has been for supporting the education and inclusion of hearing impaired students in schools and classrooms by providing transcriptions of speech [23, 72, 73, 74, 75, 76, 77]. Stewart and McKee investigated using speech recognition technology in practice to support deaf higher education students in lectures already in 2003 [72]. Their students relied primarily on sign language interpreters, which presented many practical issues. Firstly, the availability of sign language interpreters caused problems, as in many cases they had to be booked months in advance. Even when interpreters were available, the problem they encountered in the higher education setting was that interpreters had major difficulties translating complex technical and scientific vocabulary. As an example of this, the authors give the phrase "isochronous data transmission over Firewire or as it is also known, IEEE1394", which an interpreter could not translate at all. Furthermore, the international student body caused additional complications, as the sign languages used by people from different countries are not identical. For example, the British Sign Language and the American Sign Language differ quite significantly from each other, even though both nations use English in verbal communication. As a solution, the authors tried adapting a commercial speech recognizer (IBM Via Voice), which was reportedly considered the best commercially available continuous speech recognizer at the time. They found that while the concept was valid, deaf students liked the approach, and that it could improve the comprehensibility of a lecture, in general, speech recognition technology was not ready for the challenge.

Other previous proposals for ASR-based assistive applications for communication include the following: In 2006, Matthews et al. presented a mobile sound transcription tool using offline speech and acoustic event recognition [78]. When the user pressed a button in the application, the last 30 seconds of sound were uploaded to a server for transcription, and send back as a text message. The message included dialog and descriptions of environmental sounds. Gelder et al. described a transcription table design, a table prototype with integrated displays for each person for providing text support during meetings [79]. Transcriptions are provided for each person around the table in order not to stigmatize hearing impaired meeting participants. Lee et al. described a mobile conversational assistance system that uses acoustic beamforming with a multi-channel microphone array for suppressing background noise [80]. The beamforming array is used to pick up sound from the direction of the speaker, while attenuating sound from other directions.

Mirzaie et al. used the *audio-visual speech recognition* (AVSR) technique for accurate speech-to-text translation even in noisy environments [24]. Additionally, the results were displayed utilizing *augmented reality* (AR), where the text was displayed floating next to the speaker. The AVSR system combines audio-based speech recognition with computer vision detection of the mouth and facial expressions of the speaker. A visual feature observation set is produced from consecutive video frames of the tracked mouth region of the speaker. The audio and video features are processed jointly to produce the speech recognition result. Adding the vision-based component can help the recognition accuracy especially in noisy situations. However, the AVSR system used was an offline recognizer with an average processing time of ten seconds, which limits the systems usability for real-time speech recognition.

Kushalnagar et al. investigated enhancing the accessibility and readability of real-time speech-to-text display in classroom settings [81]. Typical implementations of speech-to-text displays can still present subtle , but noticeable difficulties for deaf and hard of hearing students. These issues do not depend on the speech-to-text method used, thus applying to both human transcribers and ASR systems. According to Kushalnagar et al., one key issue is that watching the text display distracts and takes time away from following the teacher and the visuals presented. Hearing students can observe the visual information provided, such as charts and images, while at the same time listening to the verbal description, whereas hearing impaired students have alternate between the two. The term *visual dispersion* is used for the juggling between of multiple concurrent visuals, and has been offered as a major reason why deaf and hard of hearing students get less out of lectures than their hearing peers. Falling behind in reading the text can be a problem as well. Deaf students with sign language as their primary language can be less fluent in English as their hearing peers, and consequently, read text and captions more slowly. To address these problems, a *tracked speech-to-text display* (TSD) method is presented. The goal of the TSD system is to minimize the distance between the text display and the speaker. This is accomplished by tracking the position of the teacher, and using a video projector to display text next to the teacher. In effect, TSD implements

an analog AR environment, where the added object is projected directly to the environments instead of being virtually added through a display screen. Microsoft Kinect was used for the motion tracking. The results from a real-world classroom comparison between regular speech-to-text display and the tracked method indicated that both hearing impaired and normal hearing students preferred the tracked text display, and that it helped to follow the lecture.

There have also been assistive solutions that are not based on automatic speech recognition, but fulfill a similar role nonetheless. Bragg et al. presented a mobile sound detector application for deaf and hard of hearing poeple [82]. The app provides personalized sound alerts of sounds other than speech, such as alarms, sirens and doorbells. The user can customize the application to work with the sound events they are interested in. For recognizing these sounds, *acoustic event detection* (AED) is used. AED is technically very similar to ASR, as MFCCs, HMMs and GMMs are commonly used for recognition. The app alerts the user with vibration and pop-up notifications when it detects a pre-determined sound event that the user has previously recorded and defined. Similarly, using a smartwatch for environmental sound alerts showed promise from brief testing with six deaf participants [83].

One common variation for an assistive application for the hearing impaired has been to simply communicate using written text and symbols. In essence, these applications are the digital version of a pen and paper for writing messages or cards and booklets traditionally utilized by the deaf. Hirayama presented such a communication aid software for visual communication [84]. A mobile application was implemented that can display frequently used sentences in everyday life or in emergency communication. Images and visual objects can be used as well for various functions. For example, a *SOS card* was implemented, which can be used in emergency situations to visually indicate the source and type of pain. First, the hearing impaired user selects a sentence or image from a list on the smartphone screen. The message is then conveyed by showing the display to other persons. The advantages of this type of software application come mainly from replacing old, analog implementations of the same principle, with the software being faster and more versatile.

Utilizing automatic speech recognition to augment human written language translators has been researched as well: Gaur et al. [32] investigated using ASR output as a starting point for human transcription, with the goal of improving the latency of results produced by human transcribers. They found that the effectiveness of this approach depends highly on the quality of the ASR output: Accurate ASR could improve the efficiency of human transcription, but when the WER was 30% or more, humans were better off starting from scratch.

Relating to designing and implementing assistive applications, the general requirements and architectures for applications developed specifically for the hearing impaired have been investigated: Mielke et al. presented a detailed analysis of the

needs of hearing impaired people in the context of an acoustic event detection application. Requirements for a suitable assistive application were derived from the results of the analysis, and used to define an architecture for the implementation of such a system [43]. They propose the use of environmental sound recognition algorithms to help individuals with severe hearing impairment perceive acoustic information, such as road traffic. As general requirements and desired properties, for example the following are presented: Assistive devices should be small and sleek, and preferably look like mainstream devices in order not to expose and underline the impairment. Devices and applications should have low power consumption so that they remain usable for a full day. Devices should integrate easily to the daily life of the user. The user interface should be comfortable and easy to use. Special attention should be paid to the privacy of uploaded data, especially sound files that can contain sensitive information of the user and other people in the same environment. Smartphones were found to be a good platform for implementing assistive solutions, due to the processing power available in them, combined with the fact that they are widely used by most people and 96% of deaf individuals as reported in one study [43]. Another advantageous feature of smartphones is the direct internet access.

In an article published in 2015, Prietch et al. conducted a systematic review of the literature on speech-to-text applications for deaf and hard of hearing individuals, and other related work [85]. The review was undertaken in order to elicit application requirements for mobile applications using a speech-to-text system. They found that in previous research, the problems concerning the quality of transcriptions were emphasized. The latency of transcription was indicated as a challenge, as well as the difficulty in understanding automatic transcriptions due to the lack of punctuation marks (e.g., commas and periods). In general, positive results from the use of speech-to-text services in education were observed. One study reported that deaf and hard of hearing students receiving text transcripts achieved better grades than students receiving instruction with interpreters only. Reading proficiency is considered to be important for good information retention. Overall, the acceptance of speech-to-text systems was found to be linked to the quality of the transcriptions. For the requirements for speech-to-text applications for use in inclusive classrooms, the following were presented, among others: Identify the person who is speaking, save texts for later reading, record date and time of conversation, ability to turn audio input off when desired, be available 24 hours a day, and adjustable font size.

3 Conversation Assistant

This section describes the proposed Conversation Assistant solution and presents the detailed information on the implementation of the developed software prototype, as well as the reasoning behind the choices made. The Conversation Assistant can be best viewed as a general approach for supporting communication for deaf and hard of hearing persons, as it is not tied to a single possible implementation method or hardware device.

3.1 Description

The basic operation principle of the proposed Conversation Assistant can be described as follows: First, the speech of a speaker is picked up with a microphone. This can be for example the build-in microphone of a users device, or an external microphone, preferably close to the speaker for improved signal-to-noise ratio and ASR performance. The speech input is then converted into text by an automatic speech recognition system. Finally, the recognition result is displayed on a screen for the user or users. All of these basic elements can be separate from each other, or integrated into a single device. The resulting text can be displayed with special visual formatting depending on additional features the system has, such as speaker diarization, which would enable visual separation of speech from different speakers. A hearing impaired individual can then use the text transcription for visually acquiring the spoken communication as an alternative to sound and hearing.

The Conversation Assistant could support conversations and communication in two ways, depending on the level of hearing loss: In the case of a deaf individual, it could enable following spoken communication without the need for a human translator or other special arrangements. At the same time, it could enable normally hearing people to more easily and spontaneously communicate to deaf people, as they can simply use speech without having to go through an intermediary. For two-way communication between the deaf and hearing, the process can be reversed: a speech synthesizer could be used to translate a written answer from the deaf person back to speech. The speech synthesizer element could optionally be integrated into the Conversation Assistant application directly, but that additional feature is not investigated in this work. For the hard of hearing, the Conversation Assistant could be used concurrently with listening, with the Conversation Assistant as backup and support for words and moments they did not hear clearly. In this capacity, the Conversation Assistant could be useful especially in acoustically challenging situations and environments, where hearing aids and cochlear implants typically struggle. As their speech perception declines, they can rely more on the text transcription. Situations with a lot of background noise and clamor are the primary example of these circumstances. The one obvious problem with this objective for the Conversation Assistant is that as described in section 2.2.7, automatic speech recognition has traditionally also struggled in exactly the same types of situations. Overall, sufficiently accurate speech recognition is critical in order for the Conversation Assistant to be a viable

solution and work as intended. Consequently, sufficient accuracy is one of the key factors being investigated in the user testing, as is described in section 4.

3.2 Implementation

Moving on to the realization of the Conversation Assistant, there are a few different options for the platform and type of implementation. Ideally, the Conversation Assistant would be universally available on every device or system that meets the basic requirements: a microphone for capturing audio, the ability to run a speech recognizer, and a screen for displaying the text. Modern smart devices offer conveniently exactly such a platform. However, speech recognition, especially LVCSR, has traditionally been a computationally intensive task, and large models can require a lot of disk space and system memory during operation. As such, more powerful laptop computers can run a high-end large vocabulary speech recognizer locally, meaning on the end-devices own computing hardware, but mobile devices and less powerful computers generally cannot. Also, it would not be practical or sensible, that every user would need to download potentially multiple gigabytes of data for large models to their device in order for the speech recognition to happen locally on the device, even if the computational resources were sufficient.

Instead of doing speech recognition locally on the device, a server-based solution can be used, where the audio signal is acquired at the end-device and is then send to a server handling the speech recognition process. The server simply returns the recognition results as a text string to the device for display. This is the way most ASR systems aimed for consumers currently work already. Another option would be to do feature extraction at the end-device, and then send only the feature vectors to the server. This approach would require a lot less bandwidth, improving the latency of the system in cases where the internet connection speed is the bottleneck. The major downside of the server-based implementation is the required internet connection. Figure 12 presents an overview of the main elements of the Conversation Assistant system.

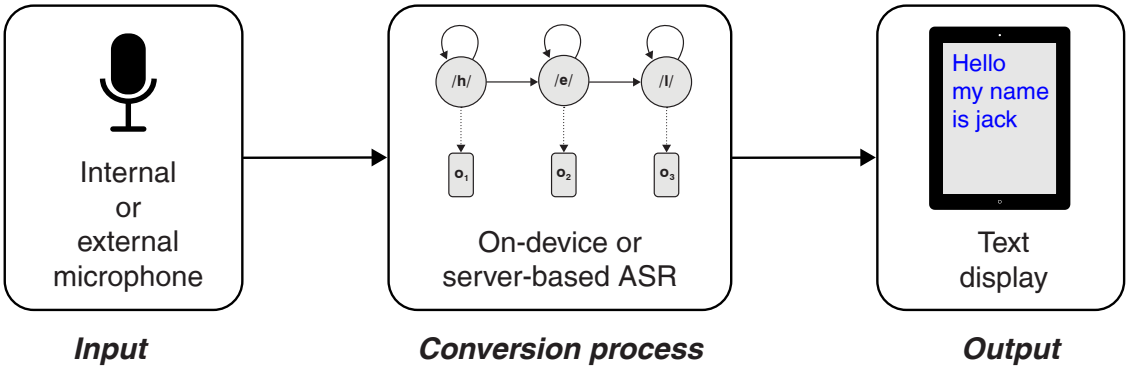


Figure 12: *The basic elements of the Conversation Assistant system.*

It should be mentioned that the are some exceptions to running an ASR system on a smartphone. Engineers from Google Inc. published a paper in 2016, where a highly-compressed and optimized large vocabulary speech recognizer for English was shown to run in real-time on a Nexus 5 Android smartphone [21]. The size of the vocabulary was 64K words, and the median speed for recognition was seven times faster than real-time. A large memory footprint reduction was achieved with SVD-based[3] compression and neural network quantization, where the floating-point model parameters, such as the weights, were quantized to 8-bit integers. However, computationally heavier and server-based recognizers remain generally more accurate [34], and arguably more practical, at least from a development perspective.

For the implementation of Conversation Assistant interface on the end-device, there are multiple possible choices: For laptops and desktop computers, it could be implemented as a desktop software application, and similarly for mobile devices as a mobile application. For both platforms, the ASR component could be either on-device or the server-based version. However, for easy universal access and platform independent implementation, a web service approach would arguably be the most sensible choice. In a web service implementation, the Conversation Assistant would be accessed through a web-page, enabling simple usage on every conceivable device. Using responsive web design, the user interface can be adapted to work on all screen In the next section, the implementation of our prototype used for user testing is described in detail.

3.3 Prototype

The primary function of the Conversation Assistant prototype was the validation of the proposed method for assisting deaf and hard of hearing persons in conversational situations. It was not intended to be a finalized or commercially ready application at this stage of the project. Instead, the prototype was only meant for use in the first round of user testing. Therefore, the implementation contained only features necessary for realizing the first round of user testing. As a consequence, the graphical user interface was left as simple as possible in order to minimize the possibility of it interfering with testing the method itself. Existing open-source software and speech recognition systems were leveraged to get quickly started with the testing process, fitting consistently with the overall theme of applying existing technology into practice that this work is founded upon. Our prototype is intended specifically for Finnish speech recognition, though the Conversation Assistant approach itself can be applied to virtually any language, as long as an ASR system is available for that language. In the case of our prototype, the language of the speech recognizer could be changed simply by changing the models and lexicon. The *Kaldi* toolkit is used for doing speech recognition. The prototype was implemented in the *Linux* operating system environment, relying on open-source software tools and frameworks. The source code for the prototype is included in appendix A.

[3] SVD, short for *singular-value decomposition*, is a matrix factorization method.

3.3.1 Kaldi ASR toolkit

Kaldi is an open-source speech recognition toolkit, intended primarily for speech recognition research [86]. Kaldi is written in C++ and licensed under the Apache License version 2.0. The goal of the Kaldi project is to have modern and flexible code, that can be easily modified and extended. Kaldi's speech recognition system is based on finite-state transducers. For tight integration with FSTs, Kaldi includes the OpenFst toolkit as a library. Extensive linear algebra support is included by wrapping the BLAS (Basic Linear Algebra Subprograms) and LAPACK (Linear Algebra Package) libraries.

Our prototype utilizes Gst-Kaldi, which is a GStreamer plugin implemented around Kaldi's online neural network decoder [52]. GStreamer itself is an open-source multimedia framework. In GStreamer, modular media-processing components are joined together into a pipeline to achieve a desired function, such as media playback, recording, transcoding, streaming and editing. The pipeline-based architecture, together with an extensive collection of processing elements and plugins enable flexible media handling and routing from a large variety of sources and formats. In our use case, audio can be inputted versatilely from soundcards, read from an audio or video file, or even captured directly from an internet media stream. Desired audio pre-processing steps like sampling rate conversion, level normalization and compression can be easily added to the GStreamer pipeline. A open source speech recognition server implementation based on the same Kaldi GStreamer plugin is also available, meaning it would be relatively simple to transition to a server-based recognition system [87].

3.3.2 Models

For the first user test prototype, previously existing ASR models were utilized instead of training new models specifically for this purpose. As speech recognition methods, and the Kaldi toolkit correspondingly, are continuously developed further, the models used did not necessarily represent the very latest and best possible accuracy in Finnish ASR. The older models were nevertheless deemed to be sufficient for our main purpose, which was to test and validate the Conversation Assistant method. First spending many weeks or months in order to harness the latest advances in ASR would be in vain, if during the user testing it becomes immediately apparent that the Conversation Assistant is not a good solution. Thus, it was preferable to get to the actual testing as fast as possible, enabled by using already existing models.

The models used in our prototype were previously developed at Aalto University's Speech Group for Finnish LVCSR, using Kaldi's nnet2 setup [88]. For the feature type, high resolution MFCCs are used with a 25 ms frame, shifted 10 ms at a time. In addition to MFCC features, Kaldi's neural network based online decoder uses *i-vectors* as an input. I-vectors are used for speaker adaptation [34, 86]. An i-vector is a vector with a dimension of several hundred, representing the speaker properties.

The DNN acoustic model has been trained using data from the Finnish SPEECON corpus, which is part of the international SPEECON speech database [89]. The corpus contains 550 adult speakers (273 males, 277 females), recorded with four microphones in four different environments, uttering a variety of prompted word sequences (e.g., names, numbers, dates, questions, single digits) and free, spontaneous speech [44, p. 43]. For the training data, the headset, lapel and far-field (one meter distance) microphone channels were used from the "public place" and "car" recording environments. The realistic noise environments containing varying levels of background noise, together with the different microphone locations are well suited for producing a more noise robust model. These recordings should also match quite well to the supposed locations, where the Conversation Assistant would be used. The language model is based on text corpora from *Kielipankki*, the Language Bank of Finland, which is a service for providing natural language resources to research usage. The specific text corpus used contained translated texts from the European Parliament. The vocabulary is morph-based.

3.3.3 Application

The prototype application is based on the simple Python GUI demo included with Gst-Kaldi. It is a desktop application and the speech recognizer is run locally, which was the most practical solution for the user testing. Extending the existing open-source demo application to fill the needs of this project enabled us to quickly start the testing process, instead of having to reinvent the wheel with a solution build from the ground up. For the Conversation Assistant prototype, the GUI was modified,

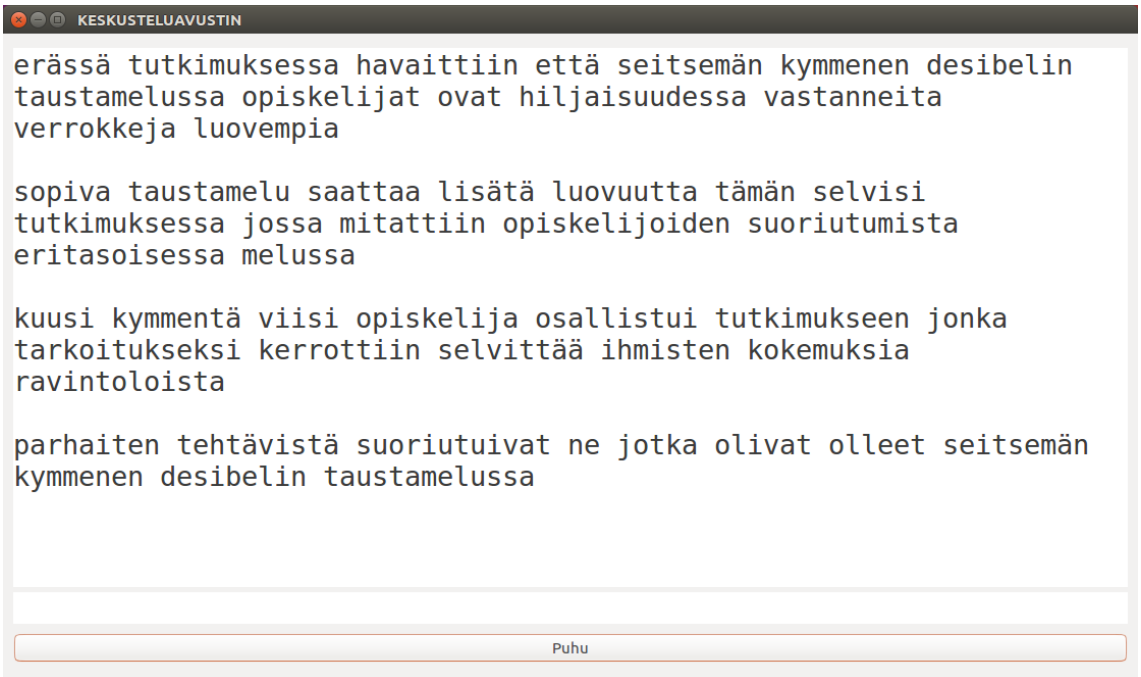


Figure 13: *The Conversation Assistant prototype application in use.*

correct text parsing of the output was implemented, and the models and parameters were updated with our own, on top of the demo application. Figure 13 presents an image of the Conversation Assistant prototype. Our simple GUI implementation consists of a single button to start and stop the speech recognizer, a one line text field for displaying the real-time recognition result, and a scrollable history window where each complete sentence is displayed in chronological order, always showing the latest results.

The application uses version three of the Python programming language. The PyGTK API for using GTK+ with Python is used for the GUI. GTK+, originally known as the *GIMP Toolkit*, is a multi-platform, open-source toolkit for creating graphical user interfaces. GStreamer and the Kaldi GStreamer plugin are used through the PyGst Python binding. In GStreamer, the default audio input of the OS is used, meaning that an external soundcard and microphone will work automatically just by setting it as the default audio device from the system sound settings, instead of having to configure it manually in the GStreamer pipeline. The application functions in the following way: A GStreamer pipeline captures audio from the Pulse Audio sound server, the sound interface system used in Ubuntu and many other Linux distributions. The audio signal is processed and routed to Kaldi's online nnet-2 decoder. The decoder has two return functions: one returns the partial recognition result one word at a time, and the other returns the complete recognition result after an utterance has been deemed to have ended. The results have to be then parsed to remove the morph boundary markers present. The partial recognition result is constantly updated on the screen, providing real-time speech recognition display appearing one word at a time. Figure 14 presents the software architecture of the prototype. The prototype was developed and used in a Linux environment running Ubuntu 16.04, though Windows and MacOS are also technically supported by Kaldi, GStreamer, and GTK+.

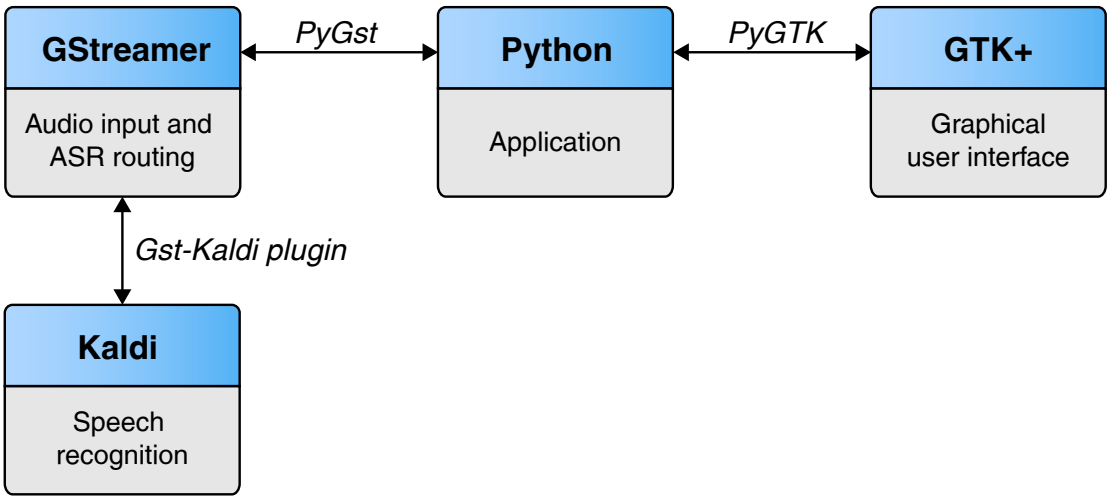


Figure 14: *The Conversation Assistant prototype software architecture.*

4 User Testing

User testing was used to validate the Conversation Assistant approach for helping with the conversational challenges faced by hearing impaired persons, and to examine how well the prototype succeeds in this, as well as to find out the key areas for improvement. Additionally, the user tests were used to gauge the interest of potential end-users towards this type of assistive approach and its commercial potential. Through user testing, it was also possible to collect valuable feedback and opinions from real end-users on for example what kind of features would they want to have, and in what kind of situations would they use the Conversation Assistant. With the information learned from user testing, it is possible to develop the Conversation Assistant to better fulfill the needs of the target audience, making it more helpful to them. Not only will this help hearing impaired people lead better lives, but can also increase the commercial potential of this type of assistive application. More commercial potential will in turn increase the likelihood that applications actually become available to the general public and in many different languages.

4.1 Objectives

The primary goal of the user testing was to determine if the Conversation Assistant is in fact helpful, and to assess how useful it is in the opinion of intended end-users. The secondary goal was to identify which factors contribute the most towards increased usefulness for the end-users. The reasoning behind this secondary objective was that while it is easy to identify many technical aspects that could be improved, like for example the speed of the speech recognizer, it might not actually matter for the end-users. If for example the users feel that on average, the speech recognition accuracy is already good enough, then improving it further would not probably increase the experienced usefulness very much. Therefore, the user tests were designed to focus more on qualitative properties as experienced by the users, instead of quantitative, objective measurements for the performance of the Conversation Assistant. It is also a lot harder to measure meaningful objective data of users interacting with devices and software, than it is to get their subjective opinion on the matter.

Ultimately, it is the subjective experience of the end users that matters when evaluating the validity and usefulness of the Conversation Assistant. In fact, typical objective measurements like the word error rate of the speech recognizer might not actually be the most relevant for this specific application [90]. A better word error rate does not automatically mean that the quality of the recognition result is also better in terms of understanding the contents. Also, the end-goal of the Conversation Assistant is not to have perfect transcriptions of speech, but to support and supplement the users hearing in real-time conversational situations. If the user can understand everything spoken to them with the help of the system, even though the text is full of errors, then it is succeeding in its intended purpose regardless of the speech recognition accuracy. Therefore, in real-time situations the type of errors is much more important than simply the overall percentage when it comes to

understandability and helpfulness [90]. This observation feels inherently logical for conversations: It is easy to imagine that a few extra letters or words, or a slightly wrong grammatical case might not affect the understandability of a sentence very considerably, whereas a missing key word might render the transcript almost useless. In conversational situations between humans, a lot of information can be interpreted based on the context and other factors like non-verbal cues (**V!**). This was reported also by our test users, saying that in many cases they could guess the meaning of a sentence well enough from incomplete and erroneous transcripts. Based on these observations, it could be more meaningful to quantify what kind of errors the speech recognizer makes, and more importantly, how each type of error affects the usefulness of the Conversation Assistant.

4.2 Design

Now that the objectives are defined and the questions to which answers are wanted to are known, the question becomes how to actually answer those questions: What tests should be performed, and how should they be performed in order for the results to be valid and as accurate as possible. The latter part is especially important when the tests involve humans, as there are many complex factors involved that could introduce bias or otherwise distort the results if the tests are not designed and executed properly. There are also ethical matters and privacy concerns to consider whenever human test subjects are involved. [66]

Potential objective measurements considered included eye-tracking, which could have been used to accurately determine the percentage of time the users spend watching the Conversation Assistant screen instead of the person speaking. Eye-tracking could also be used to assess how people tend to use the Conversation Assistant: Do they look back-and-forth quickly between the speaker and the screen, or do they first try to listen a full sentence watching the speaker, and only after that check the screen for help. The User Interfaces research group at Aalto has experience in eye-tracking and could have provided the necessary equipment if desired. While this information would be interesting from a HCI point-of-view, and have some implications for the UI design and recognition result displaying, it does not answer significantly to the primary and secondary objectives set out for the tests at this stage. It is also possible to obtain a decent approximation for this data by analysing the video recording of the test session, which can also show were the test subject was looking at on any given moment during the test. To avoid manually gathering this data from the video, computer vision methods for eye-tracking could be used to get quite accurate results automatically. For example, Krafka et al. used deep convolutional neural networks for successfully tracking eye movement on mobile device cameras [91].

Using recordings for a consistent, standardized source of speech. Missing the human element and non-verbal cues. Wanted to test in real situations.

4.3 Test plan

The final test design consisted of two separate test sections and was designed to take approximately one hour in total. One hour was chosen as it represented a good balance between the test subject getting a thorough experience without test fatigue setting in. It was also a practical choice as it made scheduling the test sessions easy. Each section had first a short practice run without using the Conversation Assistant. This was done to familiarize the test subject with the task and to give them a baseline reference for their ability to understand speech, in order for them to then be able to better estimate the usefulness of the Conversation Assistant. The first test section was designed to be a passive situation for the test subject, where the test subject is only listening to speech. The second section was an active interaction situation, where the test subject is speaking and listening in equal parts. The full test routine with the schedule is presented below, followed by a detailed description of each section:

1. **Introduction** (10 min)
    1.1. Research overview
    1.2. Legal documents
    1.3. Instructions
    1.4. Background information questionnaire
2. **Section 1: Word explaining** (20 min)
    2.1. Without the Conversation Assistant (5 min)
    2.2. With the Conversation Assistant (10 min)
    2.3. Questionnaire (5 min)
3. **Section 2: Conversation** (20 min)
    3.1. Without the Conversation Assistant (5 min)
    3.2. With the Conversation Assistant (10 min)
    3.3. Questionnaire (5 min)
4. **Debriefing** (10 min)
    4.1. Questionnaire (5 min)
    4.2. Reward

4.3.1 Introduction

The user test session started with an introduction section to gently prepare the test subject for the test situation, as is the recommended standard procedure in user and usability testing [65, 66]. To begin with, the research topic and purpose of the test is explained to the participant together with their role in it. After the general orientation, the person is asked to fill the required legal documents, which in this case consisted of the typical research consent form as well as a permission to record and use the audiovisual recordings made during the session. Then the subject was presented with the test routine and schedule for the session, and more detailed instructions for their task in each section. All of the above mentioned materials were provided in written form to ensure that everything is understood regardless of the subjects level of hearing loss. Finally, the test subject was asked if they have any questions about the test or their task, and the test proceeded forward to the last part of the introduction after the subject indicated that all was clear.

At the end of the introduction section, the test subject was asked to fill the first page of the test questionnaire, which was used to gather some basic background information on the person, as well as their previous experience with automatic speech recognition technology and mobile devices. These questions included the subjects age, their current employment status (student, employed, unemployed, retired), hearing aids and other assistive devices they use, and whether they own a smartphone and/or a tablet. This information can be useful in analyzing and understanding the results, if for example the subjects age or self-professed skill with computers affected the perceived usefulness and ease of use of the Conversation Assistant.

4.3.2 Section 1: Word explaining

Section one consisted of a word explaining exercise, where the test subjects tries to deduce the word under question based on the explanation of the test administer. The words were kept fairly simple as the intention was only to confirm that the test subject had understood what had been said, not to test the deduction skills and general knowledge of the person. The word list used contained a little over 30 words, of which approximately ten were used for the first try without the Conversation Assistant, and the rest with the Conversation Assistant. The same words were used for all test subject, though not in the same exact order, to minimize their effect on the results. Section one would conclude when there was no more words left, or if the time allocated was exceeded by more than a few minutes. After the test, the test subject was asked to fill a questionnaire related to the section, asking the person's opinion on the performance of the Conversation Assistant in that particular section.

4.3.3 Section 2: Conversation

Section two was a bilateral conversation situation, which closely resembled a typical free conversation between two persons. The test administer had a list of simple conversation topics that anybody should be able to talk about comfortably, like food,

traveling, entertainment, and hobbies. The topics were presented in the form of a question, like for example "what are your favorite foods?" or "what would you do if you won the lottery?". To initiate the conversation, the test administer would start with a question like this, and then wait for the test subject to answer it, before answering himself. Then, the test administer would keep the conversation going by asking follow-up questions and continuing to discuss the topic. After a topic was exhausted, the test administer would move on to the next topic and start the cycle of questions again. One topic was done without the Conversation assistant, and there rest with it. As before, there was a questionnaire in the end.

4.3.4 Debriefing

After both test sections, the test subject was asked to fill one last questionnaire about the Conversation Assistants overall usefulness. The post-test questionnaire also contained broader questions about this type of assistive application in general, like "in what situations would you use the Conversation Assistant?", as well as how much would they be willing to pay for it. The session ended with filling the financial forms for paying the reward.

4.4 Questionnaire

Questionnaire design principles. [65]. Unbiased questions.

A linear scale with discrete steps from one to seven was used for the numerical questions. Other question types included questions with binary yes or no answers, multiple choice questions, and written-answer questions. Each numerical question also had a text field for optional written comments, excluding some of the background questions as well as question with a written answer already.

The questionnaire was implemented with Google Forms, an online tool for creating for creating surveys and questionnaires. Using a web-based interactive questionnaire instead of a printed paper version had numerous benefits in addition to being fast and simple to implement: it is easy to control that all required questions are answered, and more importantly, that each type of question is answered in the correct way. For example in multiple-choice question, only one option can be picked in the web widged as was intended. Best of all, all the numerical data and text is acquired directly in digital format, formatted appropriately in a spreadsheet without the need to manually transcribe the data from a paper. Not only does this save a significant amount of time, but it also ensures that there aren't any errors in the collected data due to human transcription errors. Transcribing hand-written text can be especially problematic due to ambiguous or unreadable handwriting.

The questionnaire was originally made and administered in Finnish, as that was the language used in the Conversation Assistant prototype and also the native language of the test subjects. It has been translated here to English as accurately as possible,

but there are some small differences in the exact wording of the questions. A translation of the test questionnaire is presented below, and the actual questionnaire used in the test sessions is included in appendix B.

Background information

1. Name?

2. Email?

3. Age?

4. Occupation?

(a) *student* (b) *employed* (c) *unemployed* (d) *retired*

5. Do you own a smartphone?

(a) *yes* (b) *no*

6. Do you own a tablet?

(a) *yes* (b) *no*

7. Do you use a hearing aid or other assistive devices for hearing? If yes, please specify what.

8. Have you previously used an application or service that uses automatic speech recognition?

(a) *yes* (b) *no*

9. If you answered *yes* to question 8, please specify what applications and services?

10. In your own opinion, how proficient are you at using computers and mobile devices in everyday life?

bad *1* *2* *3* *4* *5* *6* *7* *good*

Section 1

11. Did the Conversation Assistant help you to understand speech?

not at all *1* *2* *3* *4* *5* *6* *7* *very much*

12. Was using the Conversation Assistant easy?

not at all *1* *2* *3* *4* *5* *6* *7* *very easy*

13. Did using the Conversation Assistant make it harder to follow speech?

not at all *1* *2* *3* *4* *5* *6* *7* *very much*

14. Was the speech recognition fast enough?

too slow *1* *2* *3* *4* *5* *6* *7* *fast enough*

15. Were the speech recognition results accurate enough (speech was recognized correctly)?

unusable *1* *2* *3* *4* *5* *6* *7* *good enough*

Section 2

16. Did the Conversation Assistant help you to understand speech?

not at all *1* *2* *3* *4* *5* *6* *7* *very much*

17. Was using the Conversation Assistant easy?

not at all *1* *2* *3* *4* *5* *6* *7* *very easy*

18. Did using the Conversation Assistant make it harder to follow speech?

not at all *1* *2* *3* *4* *5* *6* *7* *very much*

19. Did using the Conversation Assistant slow down the conversation?

not at all *1* *2* *3* *4* *5* *6* *7* *very much*

20. Was the speech recognition fast enough?

too slow *1* *2* *3* *4* *5* *6* *7* *fast enough*

21. Were the speech recognition results accurate enough (speech was recognized correctly)?

unusable *1* *2* *3* *4* *5* *6* *7* *good enough*

Debriefing

22. Was the Conversation Assistant useful in the test situations?

not at all *1* *2* *3* *4* *5* *6* *7* *very much*

23. Do you consider it to be important that the size and color of the font can be freely adjusted?

not at all *1* *2* *3* *4* *5* *6* *7* *very important*

24. Did using the Conversation Assistant make it harder to follow speech?

25. What was good about the Conversation Assistant?

26. What needs to be improved in the Conversation Assistant?

27. What features would you like to have in the Conversation Assistant?

28. Would you use, or have you already used an application like the Conversation Assistant?

29. In what situations would you use the Conversation Assistant?

30. How much would you be ready to pay monthly for an application like the Conversation Assistant?

(a) *0€* (b) *1-5€* (c) *5-10€* (d) *10-20€* (e) *20-30€* (f) *more than 30€*

31. How much would you be ready to pay as a single payment for an application like the Conversation Assistant?

(a) *0€* (b) *1-5€* (c) *5-10€* (d) *10-20€* (e) *20-30€* (f) *more than 30€*

32. Would you rather pay a monthly fee or a single payment for an application like the Conversation Assistant?

(a) *Single payment* (b) *Monthly fee*

4.5 Execution

The testing sessions were conducted at the Aalto university Acoustics Laboratory listening room, which is designed to conform to the strict room acoustic requirements set in the International Telecommunication Union recommendation ITU-R BS.1116 for performing critical listening tests [92, 93]. The dimensions of the room are 6,25 and 5,6 meters, for an area of 35 square-meters. The test setup consisted of a small table in the middle of the room with two chairs facing each other on opposing sides of the table. One chair was for the test subject and the other for the person administering the test. A Lenovo ThinkPad T460p 14" laptop running the Conversation Assistant application was placed on the table in front of the test subject. For audio input, the test administer had a DPA 4061 miniature microphone as a lapel microphone, connected to the laptop through a Focusrite Scarlett 2i4 USB audio interface. The purpose of the microphone was to capture the speech of the test administer, i.e., the person who the Conversation Assistant user (test subject) is conversing with. The speech of the application user (test subject) was not intended to be captured or translated to text, though some of it was nevertheless picked up by the microphone. The loudspeaker configuration in the listening room consists of nine Genelec 8260A active loudspeakers positioned evenly on a circle, at ear-level when sitting on a chair. The chair of the test subject was positioned to be approximately in the middle of the loudspeaker circle for optimal audio reproduction. The need for loudspeakers is explained in detail in the next section. The test session was recorded using a Panasonic video camera with a RØDE on-camera stereo microphone facing the test subject. The full test setup layout is presented in figure 15, and a photograph from the real situation is shown in figure 16.

4.5.1 Background noise simulation

In order for the user test situation to better reflect the actual environments and situations where the Conversation Assistant would be typically used, background noise recordings were utilized to simulate a more realistic setting. Background noise is one of the critical factors for the whole Conversation Assistant process in two ways: Firstly, noise directly affects how well people can hear and understand speech [38], and secondly, noise can drastically affect the accuracy of automatic speech recognition [44]. Furthermore, the impact of noise on speech intelligibility is generally even more pronounced for hearing impaired persons [7], which was also reported by our test subjects.

The low background noise level and distinct acoustic properties of the listening room, namely, a short reverberation time of 0,3 seconds, offer a very idealized environment for both listening to speech and automatic recognition of speech. As such, the results obtained without added background noise would in all likelihood apply poorly to actual real-life usage. During the user tests, background noise recordings proved to be essential for successful testing, as many of the test subjects could hear and understand speech remarkably well with the help of modern hearing aids and

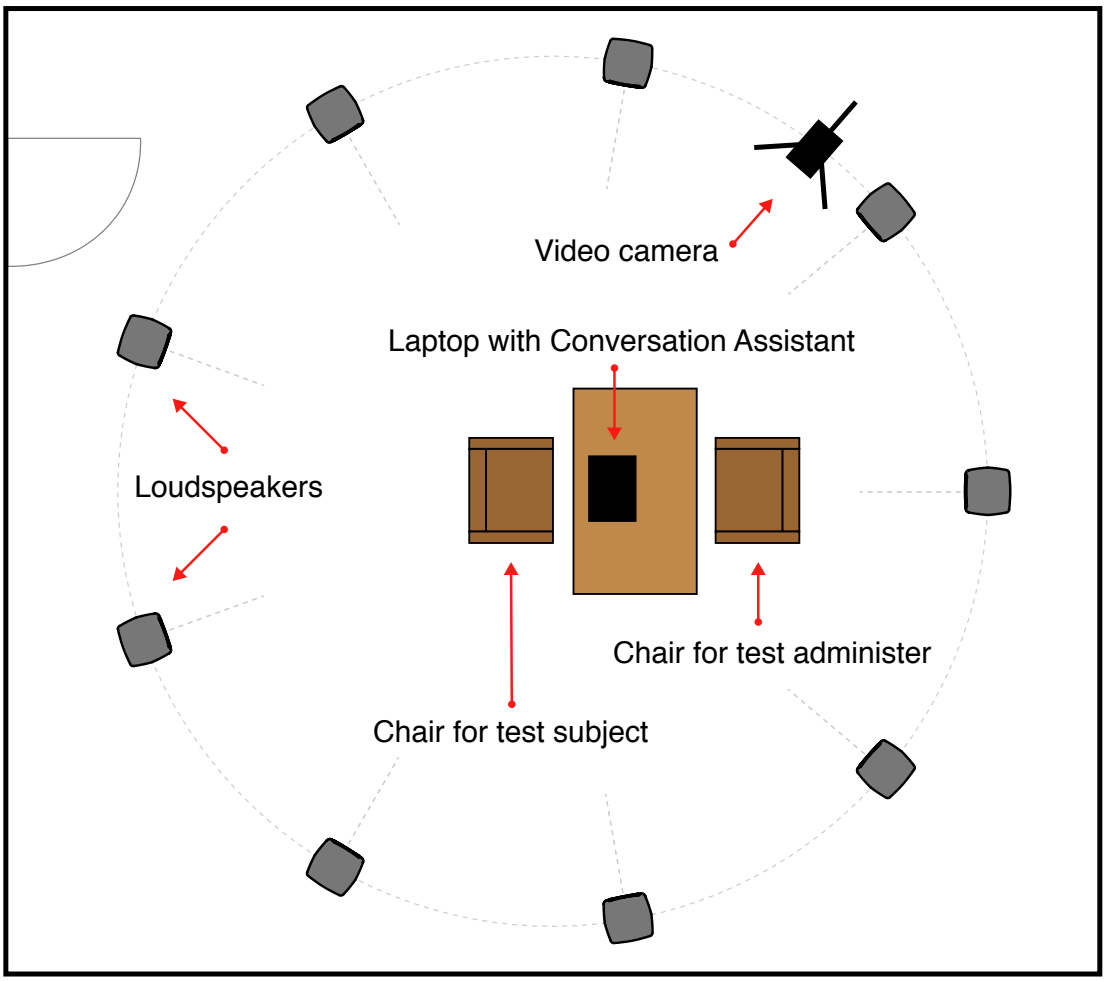


Figure 15: *User test setup in the listening room.*

cochlear implants, even though they were clinically categorized as having severe hearing loss or deafness. The background noise level was set to a predefined level at the beginning of each test section, approximately matching the sound pressure level measured at the recording location. The level was adjusted during the test session if needed, in cases where the test subject felt that they could hear too well, meaning that they didn't need to rely on the Conversation Assistant at all in order to follow the conversation.

The multi-channel audio listening setup installed in the listening room, combined with surround sound recordings in B-format enabled us to accurately and realistically reproduce the surround sound field present at the recording locations. B-format surround recordings are captured using a coincident microphone array, which produces four microphone signals: one omnidirectional (W), and three figure-of-eight channels on an orthogonal axis (X, Y, Z). These four signals describe the full-sphere sound field at the location of the microphone array, and can be decoded for playback on an

Figure 16: *A photo of the test environment.*

arbitrary loudspeaker configuration (though a minimum of four loudspeakers are needed for reproducing the horizontal plane and at least six for full-sphere sound). [94, 38, p. 287–290]

The Aalto Spatial Sound research group provided us with their previously made background noise recordings of public places, recorded using a SoundField ST350 portable surround microphone. Each of the two sections of the user test had their own noise environment. The first background is a city street containing mostly traffic noise, recorded near the Havis Amanda statue at the Helsinki Market Square (Kauppatori). The second is a busy cafe located on the Boulevard (Bulevardi) street in Helsinki, containing clamor and noise typical for busy cafes with poor acoustics, as well as some quiet background music. These two environments were selected because they represent locations were conversations often take place, the type of noise is challenging to both humans and automatic speech recognition, and the recordings were consistent in sound and level. Other recordings that could have been chosen included the inside of a moving bus and a regional train.

The Directional Audio Coding (DirAC) method developed at the Aalto University Acoustics Lab was used to decode the B-format recordings for the nine-channel speaker configuration used. The DirAC decoder divides the B-format audio input

into frequency bands using the Equivalent Rectangular Bandwidth (ERB) psychoacoustic frequency scale. For each frequency band, the B-format audio is then divided into single-channel audio channels for each loudspeaker using virtual cardioid microphones based on the loudspeaker configuration information given to the decoder. Directional and diffuseness analysis is performed for each band and used to adjust the gain and diffusion parameters of each loudspeaker channel within the frequency band. Each loudspeaker signal is then the sum of all frequency bands. [95, 38, p. 291–292]

Using DirAC, the recordings were rendered into nine-channel uncompressed PCM audio files (.wav) for easy playback. Only the horizontal sound plane was used, as it was deemed to be enough for the purpose of the user tests. Both audio scenes had multiple recordings of around one minute in length on average, made successively in the same location. For the tests, a constant background noise playing continuously during each test section was needed. Therefore, each pre-rendered nine-channel recording was split into separate mono files for each channel using MATLAB. Then the clips from each channel were edited together to form a longer loop in an audio editing software, and finally combined back to a nine-channel file in MATLAB. The Max/MSP visual programming language was used to play the resulting nine-channel PCM audio loops. The Max patch consists of a simple GUI for loading the audio file, controlling playback and adjusting the volume, which is presented in figure 17.

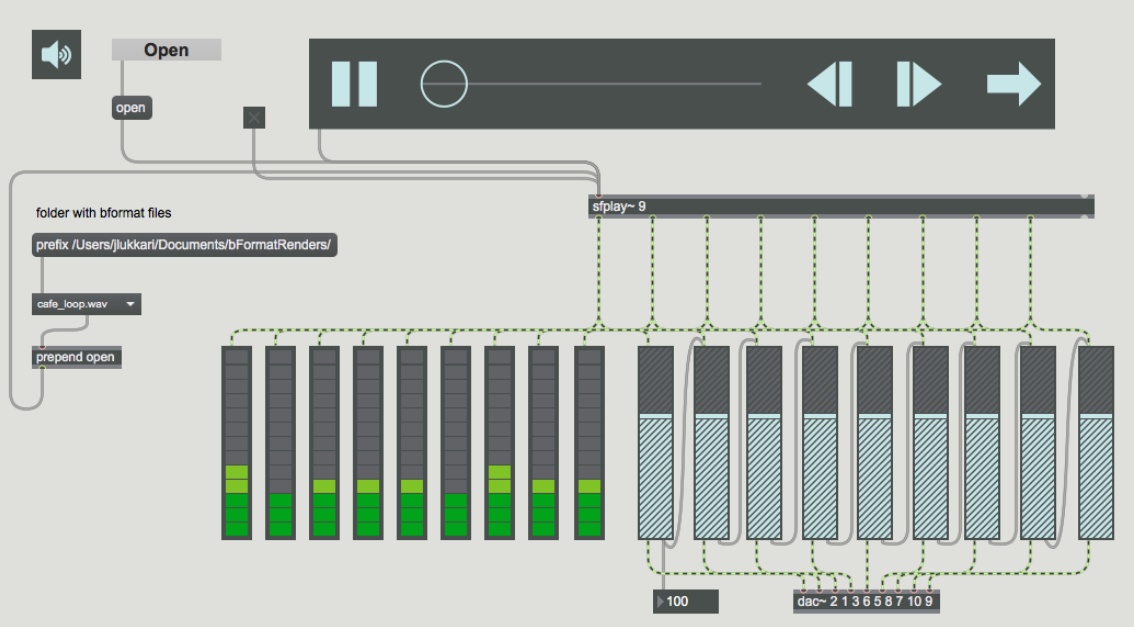


Figure 17: *Max/MSP patch for playing nine-channel audio files.*

Table 3: *Test participants.*

Sex	Age	Status	Assistive devices
female	-	retired	hearing aids in both ears
female	-	employed	none
female	15	student	cochlear implants in both ears, FM system in school
female	36	employed	none
female	38	employed	hearing aid in one ear
female	40	employed	cochlear implant, induction loop
female	50	employed	hearing aid, induction loop, FM system
female	56	employed	hearing aid in one ear, FM system
male	74	retired	cochlear implant, hearing aid

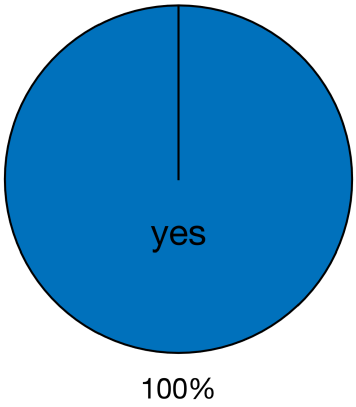
4.5.2 Test participants

The plan was for 10-12 test participants. After the preliminary filtering, we had ten suitable persons who were able to participate in the test during the days and times suitable for us. One participant had an accident that resulted in a minor injury, forcing her to cancel a few days before the scheduled test session. Unfortunately, we were not able to reschedule a new time, and as a consequence ended up with only nine test subjects in total. The relevant basic information about the test subjects is presented in table 3. There were eight females and one male test subject, ranging from the age of 15 to 74. Two of the test subjects did not report their age, but were estimated to be between 60 and 70 years old. The average age for the seven persons who reported it was 44 years, moving closer to 50 years when including the estimated age for the two others. The test participant group included one student, two retirees, and six employed persons. Two of the participants did not use any major assistive devices (cochlear implant or hearing aid), and were completely deaf. They could however speak and answer questions verbally. Three of the participants had cochlear implants, one in both ears and two in one ear. One of these two used a hearing aid in the other ear in addition to a cochlear implant. Four persons relied on a hearing aid, three in one ear and one in both ears.

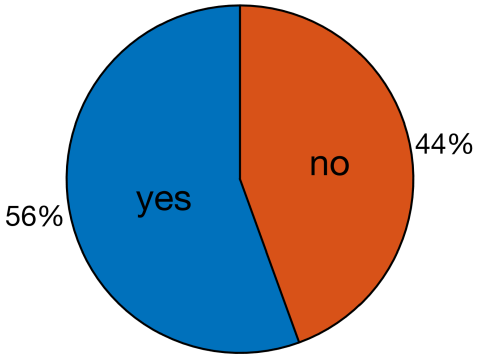
5 Results

In this section, the results obtained from the Conversation Assistant user testing are presented and analyzed. In the user testing, we focused on obtaining the subjective opinions of the intended end-users on the overall usefulness of the Conversation Assistant, as well as gathering user feedback on the key areas still in need of improvement. By design, our user testing produced qualitative data. MATLAB was used for analyzing the test data, as well as producing all figures.

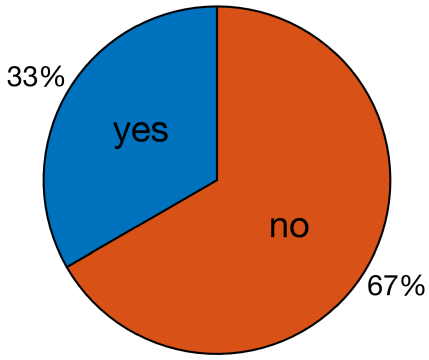
Answers to questions with a *yes or no* answer are visualized in figure 18. The first three of these questions were from the introduction section's background questionnaire (Q5, Q6, Q8), and the fourth from the debriefing section (Q32). All test



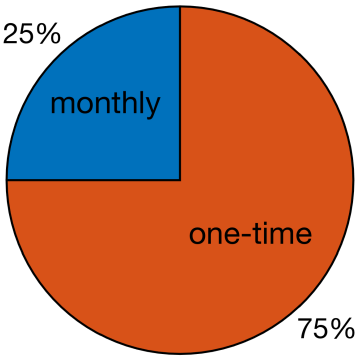
Q5: *"Do you own a smartphone?"*



Q6: *"Do you own a tablet?"*



Q8: *"Have you previously used an application or service that uses automatic speech recognition?"*



Q32: *"Would you rather pay a monthly fee or a one-time payment for an application like the Conversation Assistant?"*

Figure 18: *Binary questions from all sections.*

participants owned a smartphone, and a little over half owned a tablet (five out of nine).

5.1 Numerical ratings

The numerical data obtained, i.e., the numerical ratings given by the test participants, are presented with a box plot for each question, divided by the section. The actual numbers given by the test participants for each question are included in appendix C. In the box plots, the (red) central mark represents the median value. The bottom of the box (in blue) corresponds to the 25th percentile, and the top of the box to the 75th percentile. It should be noted that when the median is not centered in the box, it shows sample skewness, meaning the values are distributed asymmetrically. The lines (in black) extending from the box display the range of all values that are not considered outliers. A value is judged to be an outlier and marked with a red cross if it is more than 1,5 times the interquartile range (i.e., the size of the box) away from the top or bottom of the box. [96]

Figure 19 presents the data from the background questionnaire, which consists only of one question measuring the self-perceived proficiency with computers and mobile devices. The question was included in order to get a rough estimate for how much experience the participant has with these devices and how comfortable they are with technology in general. The subjective wording in the assessment served a purpose as well: Someone who feels that they are good with smart devices likely has a very positive attitude towards technology, regardless of how much knowledge and skill they might actually possess. Conversely, someone who rates their skills

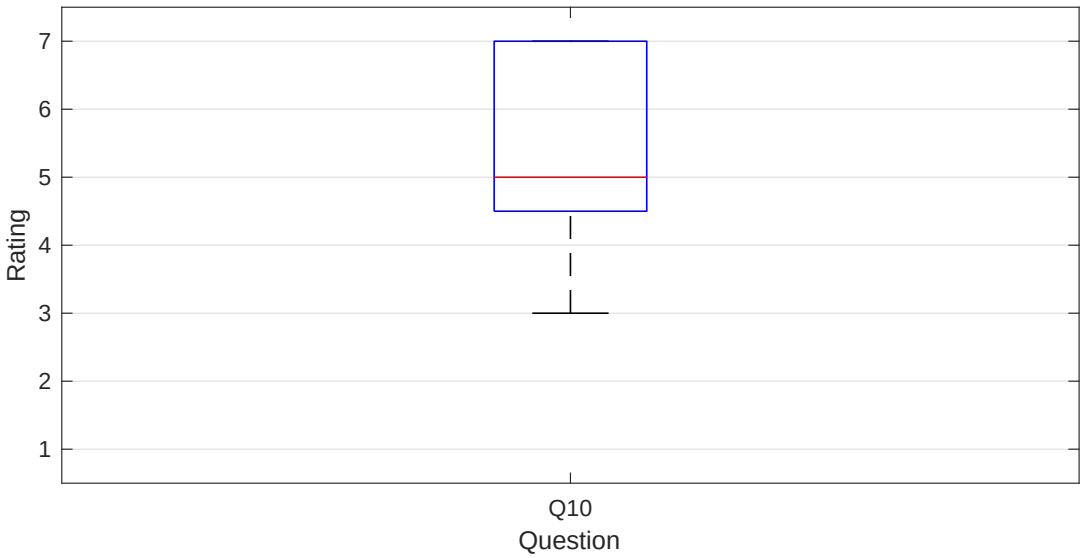


Figure 19: *Question 10: "In your own opinion, how proficient are you at using computers and mobile devices in everyday life?"*

very low likely has some aversion and negative expectations towards technology, even if their skills might actually be relatively comparable to someone with a higher rating. Therefore, the answers to this question, together with the other background questions, can be quite useful for better understanding the answers to other questions. For example, if an experienced user thinks the application is easy to use, it might not tell that much about its actual ease-of-use, especially for novices. Whereas, if someone who feels they are very bad at using computers thinks the application is very easy to use, the opinion has much more weight to it. In the same way, if a skilled participant, who owns both a smartphone and a tablet thinks the application is really hard to use, then it most probably is. If a person with a low reported proficiency thinks it's hard to use, it can be more due to their lack of experience with these devices and negative attitude, instead of any design flaws in the application itself. The question also served as a gentle introduction to the questionnaire's format and functioning, preparing and giving the participants some practice with it before moving on to the main questions.

Figure 20 presents the data from section one of the user test. The questions were the following:

Q11 Did the Conversation Assistant help you to understand speech?
(not at all – very much)

Q12 Was using the Conversation Assistant easy?
(not at all – very easy)

Q13 Did using the Conversation Assistant make it harder to follow speech?
(not at all – very much)

Q14 Was the speech recognition fast enough?
(too slow – fast enough)

Q15 Were the speech recognition results accurate enough (speech was recognized correctly)?
(unusable – good enough)

Figure 21 presents the data from section two of the user test. The questions were the following:

Q16 Did the Conversation Assistant help you to understand speech?
(not at all – very much)

Q17 Was using the Conversation Assistant easy?
(not at all – very easy)

Q18 Did using the Conversation Assistant make it harder to follow speech?
(not at all – very much)

Q19 Did using the Conversation Assistant slow down the conversation?
(not at all – very much)

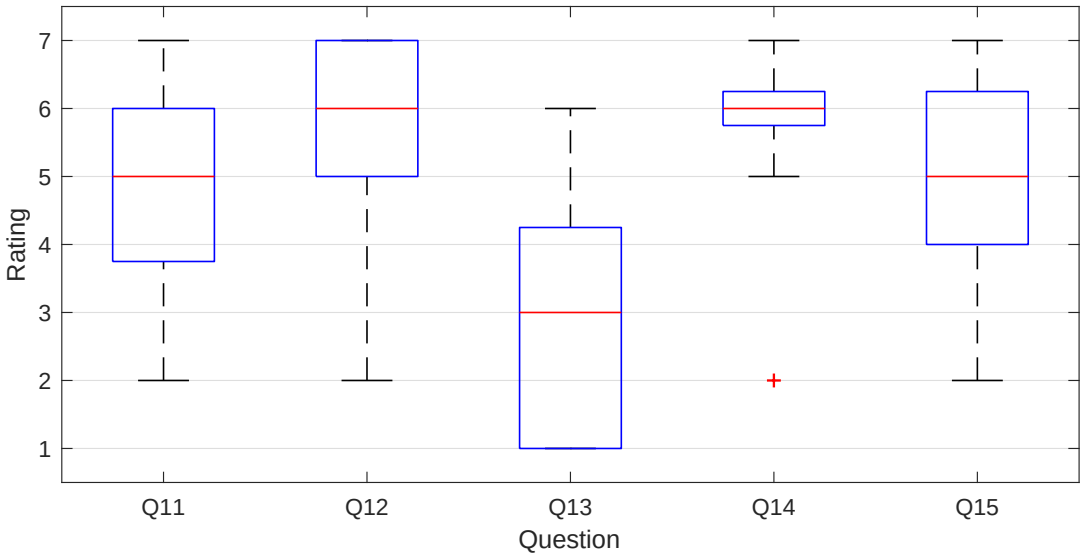


Figure 20: *Section 1 results.*

Q20 Was the speech recognition fast enough?
(too slow – fast enough)

Q21 Were the speech recognition results accurate enough (speech was recognized correctly)?
(unusable – good enough)

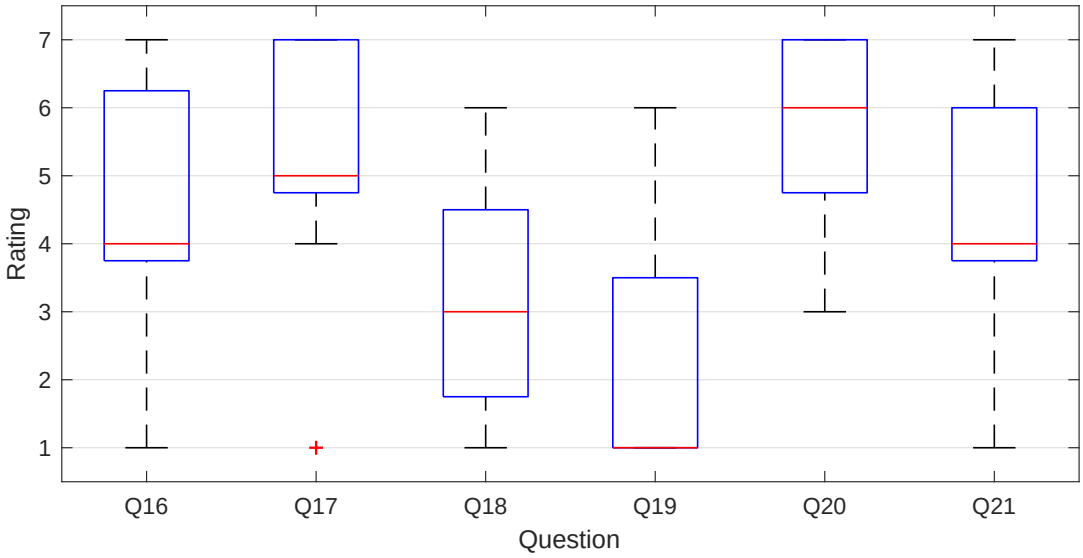


Figure 21: *Section 2 results.*

Figure 22 presents the data from the debriefing. The questions were the following:

Q22 Was the Conversation Assistant useful in test situations?
(not at all – very much)

Q23 Do you consider it to be important that the size and color of the font can be freely adjusted?
(not at all – very important)

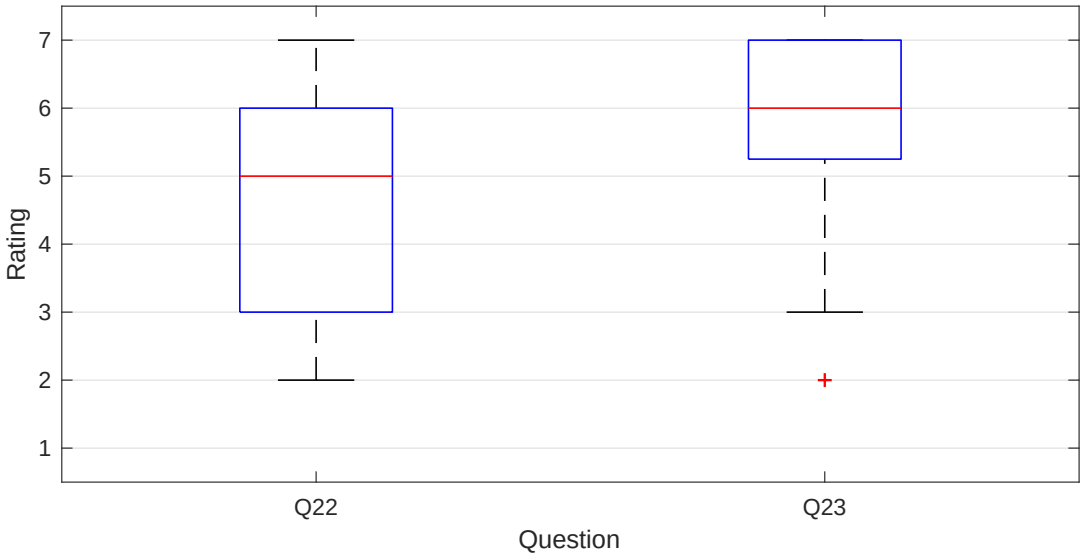


Figure 22: *Debriefing results.*

Figure 22 presents the results for how much the participants would be ready to pay monthly or as an one-time purchase for an application like the Conversation Assistant?

5.2 Verbal feedback

5.3 Analysis

One factor to keep in mind when analyzing the results is that the people who participated in the testing showed enough interest towards the Conversation Assistant to take part in the testing, meaning the test participants have to be viewed as inherently biased in some small degree towards the application. It seems probable that someone with no interest towards this type of assistive application would most probably not participate in testing it either.

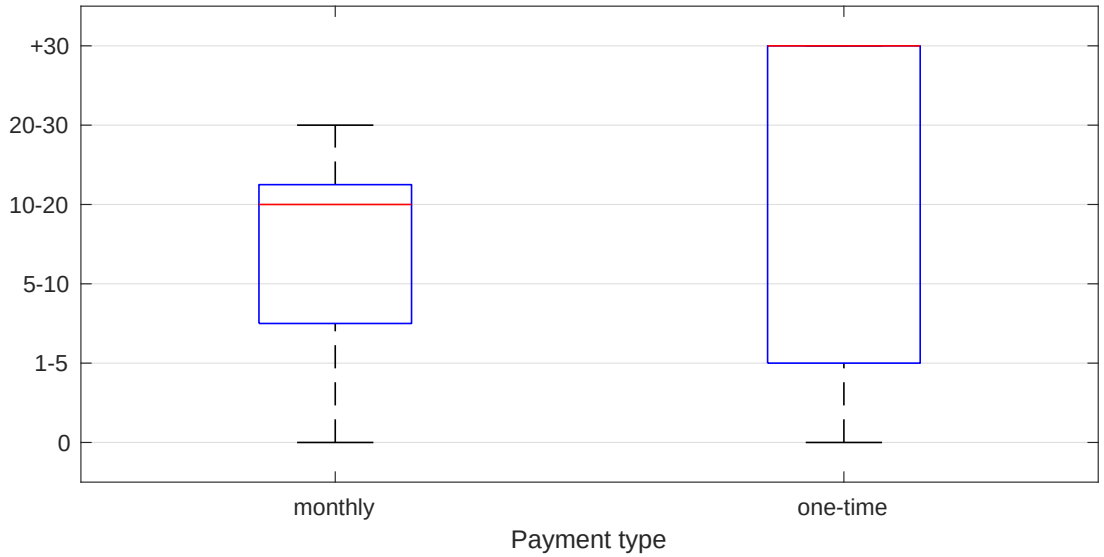


Figure 23: *"How much would you be ready to pay for an application like the Conversation Assistant?". Price in euros.*

6 Conclusions

Based on the reported substantial, and increasingly growing prevalence of hearing loss, new and affordable solutions for lessening the negative effects hearing impairment are urgently needed.

During the user testing, one of the test participants had a written language interpreter with her, as she was complete deaf and didn't use any hearing devices. This offered a good opportunity to informally compare the Conversation Assistant with a human, professional speech to text translator. The translator had a laptop computer that was placed next to the Conversation Assistant laptop on the table in front of the test user. The translator had a wireless keyboard that she used to write text to the screen, and the end result was effectively and visually pretty similar to the Conversation Assistant. During the introduction phase and between the test sections, both laptops were simultaneously displaying the translations, the other produced by our ASR system, and the other by the human translator. The human translator paused during the actual testing of course. Based on this brief empirical observation, the Conversation Assistant seemed to be noticeable faster than a human translator, but accuracy was still relatively far off from a human for conversational speech. However, it was interesting to note that the human translation results were far from perfect as well, also containing surprisingly many errors especially in the spelling. The most probable explanation seems that accurate translation has to be sacrificed for speed. All things considered, the Conversation Assistant didn't seem to be too far off from the performance level of a professional human translator. One area, where human translators are by far superior is that they can easily add relevant information to the translation, such as who is speaking (to whom), describing the mood or tone of voice, and also translate other sounds in addition to speech. While arguably a human translator is still better overall in speech to text translation for the hearing impaired, it takes many years of training and practice for a human translator to achieve these results. Also, one person cannot be in many places at the same time and requires monetary compensation for ones time and skills, meaning very few individuals can actually benefit from those translation services in the grand scheme of things. On the contrary, once an excellent ASR model is trained, it can be copied infinitely and used all the time in every place with practically no additional costs, enabling the best achieved level of automatic speech recognition for all very cheaply.

On the subject of transcription speed, online ASR is generally far superior to human transcription speed. Human conversion can require more than five times the original audio time, introducing significant latency for the results [32].

This first round of user testing also ended up being the only one. At the beginning of the project, two or more rounds of testing were envisioned originally. However, after the first round of testing, more iterations of the same test setup were deemed redundant without significant changes to the Conversation Assistant, which in turn was not possible within the scope of this thesis. Expanding the test setting from a

one-on-one conversation to a group conversation would have been the ideal next step, but unfortunately there wasn't enough time for it at this stage. Many test subjects reported that in one-on-one conversations were they can see the other person well can be quite easy to follow, even if they don't hear other person very well, by using traditional methods like lip-reading. This in turn led many of the test subjects to speculate that the Conversation Assistant would be much more useful in group conversations and in situations where the speaker is farther away or lip-reading is otherwise not possible. Examples given of the latter situations included lectures and video-conferencing meetings.

Staging a group conversation session would bring along many new complications and practical challenges compared to the relatively simple case of having just one test subject and one person running the test. There are also some unresolved technical challenges, like ensuring good quality audio input from all speakers without complicated microphone setups involving multiple microphones.

Technology is there but no wide-spread applications yet?

Better than old solutions and human interpreters? Speech recognition very fast [21].

Even if not completely solving it.

Machine learning enabling progress in all devices.

An assistive application like the presented Conversation Assistant prototype could not only improve the quality of life for hearing impaired persons and the people close to them, but also contribute towards a more inclusive society and even bring about wider socio-economic benefits to all as a consequence.

6.1 Future work

More testing. As they say: "There's no data like more data."

Group testing.

Better models for conversation speech.

How useful depending on level of hearing impairment!

Web and mobile application, server-based recognizer.

Microphone methods: access other peoples mobile devices in meetings. Beamforming and directional long range mics. Dereverberation. B-format processing.

Speaker diarization.

Augmented reality.

References

[1] B. C. Moore, *Cochlear hearing loss: physiological, psychological and technical issues*. John Wiley & Sons, New York, 2007.

[2] N. R. Peterson, D. B. Pisoni, and R. T. Miyamoto, "Cochlear implants and spoken language processing abilities: Review and assessment of the literature," *Restorative neurology and neuroscience*, vol. 28, no. 2, pp. 237–250, 2010.

[3] B. S. Wilson, D. L. Tucci, M. H. Merson, and G. M. O'Donoghue, "Global hearing health care: new findings and perspectives," *The Lancet*, 2017.

[4] B. Ohlenforst, A. A. Zekveld, E. P. Jansma, Y. Wang, G. Naylor, A. Lorens, T. Lunner, and S. E. Kramer, "Effects of hearing impairment and hearing aid amplification on listening effort: A systematic review," *Ear & Hearing*, vol. 38, pp. 267–281, 2017.

[5] P. C. Stacey, H. M. Fortnum, G. R. Barton, and A. Q. Summerfield, "Hearing-impaired children in the united kingdom: Auditory performance, communication skills, educational achievements, quality of life, and cochlear implantation," *Ear and hearing*, vol. 27, no. 2, pp. 161–186, 2006.

[6] J. Hietala and A. Lavikainen, *Huonokuuloinen – syrjäytetty työntekijä? Työympäristön asettamat esteet yhdenvertaiseen osallistumiseen*. Kuuloliitto ry, Helsinki, 2008.

[7] E. W. Healy and S. E. Yoho, "Difficulty understanding speech in noise by the hearing impaired: Underlying causes and technological solutions," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pp. 89–92, IEEE, 2016.

[8] I. Koskela, J. Ruusuvuori, P. Juvonen-Posti, N. Nevala, and P. Husman, "Kuulokojeen käyttäjät työelämässä: Monimenetelmäinen tutkimus kuulokojeen käytön esteistä ja edisteistä työelämässä," 2013.

[9] A. Lavikainen, *Huonokuuloisten ja kuurojen opiskelijoiden toisen asteen opinnoissa kohtaamat haasteet ja tuki opintojen aikana*. Kuuloliitto ry, Helsinki, 2014.

[10] K.-M. Blomberg and E. Lonka, "Sisäkorvaistutetta käyttävien aikuisten elämänlaatu," *Puhe ja kieli*, vol. 30, no. 4, pp. 233–248, 2010.

[11] P. Haatainen, "Viestintähaasteet kuurojen ja kuulevien yliopisto-opiskelijoiden keskinäisessä viestinnässä," Master's thesis, University of Jyväskylä, 2013.

[12] C. Mathers, A. Smith, and M. Concha, "Global burden of hearing loss in the year 2000," *WHO Global burden of Disease*, vol. 18, no. 4, pp. 1–30, 2000.

[13] W. J. McLean, X. Yin, L. Lu, D. R. Lenz, D. McLean, R. Langer, J. M. Karp, and A. S. Edge, "Clonal expansion of lgr5-positive cells from mammalian cochlea and high-purity generation of sensory hair cells," *Cell reports*, vol. 18, no. 8, pp. 1917–1929, 2017.

[14] P. Rainó, *Sisäkorvaistutteen saaneiden kuurojen lasten ja nuorten kielivalinnoista ja tulkkauspalvelujen tarpeesta*. Humak University of Applied Sciences, 2012.

[15] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. Springer, 2015.

[16] X. Huang, A. Acero, and H.-W. Hon, *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice Hall, 2001.

[17] M. Gales and S. Young, "The application of hidden markov models in speech recognition," *Foundations and trends in signal processing*, vol. 1, no. 3, pp. 195–304, 2008.

[18] S. Keronen, *Approaching human performance in noise robust automatic speech recognition*. PhD thesis, Aalto University, 2014.

[19] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[20] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[21] I. McGraw, R. Prabhavalkar, R. Alvarez, M. G. Arenas, K. Rao, D. Rybach, O. Alsharif, H. Sak, A. Gruenstein, F. Beaufays, *et al.*, "Personalized speech recognition on mobile devices," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 5955–5959, IEEE, 2016.

[22] J. Robison and C. Jensema, "Computer speech recognition as an assistive device for deaf and hard of hearing people," *Enhancing support services*, pp. 154–158, 1996.

[23] R. Kheir and T. Way, "Inclusion of deaf students in computer science classes using real-time speech transcription," *ACM Sigcse Bulletin*, vol. 39, no. 3, pp. 261–265, 2007.

[24] M. R. Mirzaei, S. Ghorshi, and M. Mortazavi, "Combining augmented reality and speech technologies to help deaf and hard of hearing people," in *Virtual and Augmented Reality (SVR), 2012 14th Symposium on*, pp. 174–181, IEEE, 2012.

[25] H. Levitt, "A historical perspective on digital hearing aids: how digital technology has changed modern hearing aids," *Trends in amplification*, vol. 11, no. 1, pp. 7–24, 2007.

[26] T. Goehring, X. Yang, J. J. Monaghan, and S. Bleeck, "Speech enhancement for hearing-impaired listeners using deep neural networks with auditory-model based features," in *Signal Processing Conference (EUSIPCO), 2016 24th European*, pp. 2300–2304, IEEE, 2016.

[27] L. M. Friesen, R. V. Shannon, D. Baskent, and X. Wang, "Speech recognition in noise as a function of the number of spectral channels: comparison of acoustic hearing and cochlear implants," *The Journal of the Acoustical Society of America*, vol. 110, no. 2, pp. 1150–1163, 2001.

[28] Q.-J. Fu and G. Nogaki, "Noise susceptibility of cochlear implant users: the role of spectral resolution and smearing," *Journal of the Association for Research in Otolaryngology*, vol. 6, no. 1, pp. 19–27, 2005.

[29] A. G. Srinivasan, M. Padilla, R. V. Shannon, and D. M. Landsberger, "Improving speech perception in noise with current focusing in cochlear implant users," *Hearing research*, vol. 299, pp. 29–36, 2013.

[30] J. Salonen, *Hearing impairment and tinnitus in the elderly*. PhD thesis, University of Turku, 2013. Annales Universitatis Turkuensis D 1055.

[31] P. C. A. Pereira and P. A. de Carvalho Fortes, "Communication and information barriers to health assistance for deaf patients," *American annals of the deaf*, vol. 155, no. 1, pp. 31–37, 2010.

[32] Y. Gaur, W. S. Lasecki, F. Metze, and J. P. Bigham, "The effects of automatic speech recognition quality on human transcription latency," in *Proceedings of the 13th Web for All Conference*, p. 23, ACM, 2016.

[33] J. Pylkkönen, *Towards Efficient and Robust Automatic Speech Recognition: Decoding Techniques and Discriminative Training*. PhD thesis, Aalto University, 2013.

[34] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[35] S. Enarvi, P. Smit, S. Virpioja, and M. Kurimo, "Automatic speech recognition with very large conversational finnish and estonian vocabularies," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[36] A. Cavender and R. E. Ladner, "Hearing impairments," in *Web Accessibility*, pp. 25–35, Springer, 2008.

[37] R. J. Smith, A. E. Shearer, M. S. Hildebrand, and G. Van Camp, "Deafness and hereditary hearing loss overview," in *GeneReviews*, University of Washington, Seattle, 1999, updated 2017. Online, accessed 4.9.2017. Available at https://www.ncbi.nlm.nih.gov/books/NBK1434/.

[38] V. Pulkki and M. Karjalainen, *Communication Acoustics: An Introduction to Speech, Audio and Psychoacoustics*. John Wiley & Sons, New York, 2015.

[39] ISO 226:2003, "Normal equal-loudness-level contours," standard, International Organization for Standardization, Geneva, Switzerland, 2003.

[40] A. W. Bronkhorst, "The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions," *Acta Acustica united with Acustica*, vol. 86, no. 1, pp. 117–128, 2000.

[41] N. Fink, M. Furst, and C. Muchnik, "The benefit of speech enhancement to the hearing impaired," in *Electrical and Electronics Engineers in Israel, 2008. IEEEI 2008. IEEE 25th Convention of*, pp. 130–133, IEEE, 2008.

[42] F. R. Lin, J. K. Niparko, and L. Ferrucci, "Hearing loss prevalence in the united states," *Archives of internal medicine*, vol. 171, no. 20, pp. 1851–1853, 2011.

[43] M. Mielke, A. Grünewald, and R. Brück, "An assistive technology for hearing-impaired persons: Analysis, requirements and architecture," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4702–4705, IEEE, 2013.

[44] H. Kallasjoki, *Feature Enhancement and Uncertainty Estimation for Recognition of Noisy and Reverberant Speech*. PhD thesis, Aalto University, 2016.

[45] T. Hori and A. Nakamura, "Speech recognition algorithms using weighted finite-state transducers," *Synthesis Lectures on Speech and Audio Processing*, vol. 9, no. 1, pp. 1–162, 2013.

[46] E. Arısoy, M. Kurimo, M. Saraçlar, T. Hirsimäki, J. Pylkkönen, T. Alumäe, and H. Sak, "Statistical language modeling for automatic speech recognition of agglutinative languages," in *Speech Recognition*, InTech, 2008.

[47] T. Hirsimaki, J. Pylkkonen, 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.

[48] A. Mansikkaniemi, "Continuous unsupervised topic adaptation for morph-based speech recognition," 2017.

[49] M. Kurimo, A. Puurula, E. Arisoy, V. Siivola, T. Hirsimäki, J. Pylkkönen, T. Alumäe, and M. Saraclar, "Unlimited vocabulary speech recognition for agglutinative languages," in *Proceedings of the main conference on Human*

Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, pp. 487–494, Association for Computational Linguistics, 2006.

[50] 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.

[51] P. Smit, S. Virpioja, and M. Kurimo, "Improved subword modeling for WFST-based speech recognition," in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association*, (Stockholm, Sweden), August 2017.

[52] T. Alumäe, "Full-duplex speech-to-text system for Estonian," in *Human Language Technologies – The Baltic Perspective: Proceedings of the Sixth International Conference Baltic HLT 2014*, vol. 268, (Kaunas, Lithuania), IOS Press, September 2014.

[53] T. Alumäe, "Recent improvements in estonian LVCSR," in *Spoken Language Technologies for Under-Resourced Languages*, 2014.

[54] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Springer Handbook of Speech Processing*, pp. 559–584, Springer, 2008.

[55] S. Keronen, U. Remes, K. J. Palomäki, T. Virtanen, and M. Kurimo, "Comparison of noise robust methods in large vocabulary speech recognition," in *Signal Processing Conference, 2010 18th European*, pp. 1973–1977, IEEE, 2010.

[56] A. Mansikkaniemi, P. Smit, and M. Kurimo, "Automatic construction of the finnish parliament speech corpus," in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association*, (Stockholm, Sweden), August 2017.

[57] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2263–2276, 2016.

[58] M. Richter and M. Flückiger, *User-Centred Engineering*. Springer, 2014.

[59] D. Deuff and M. Cosquer, *User-centered agile method*. John Wiley & Sons, 2013.

[60] L. E. Potter, J. Korte, and S. Nielsen, "Design with the deaf: do deaf children need their own approach when designing technology?," in *Proceedings of the 2014 conference on Interaction design and children*, pp. 249–252, ACM, 2014.

[61] A. Savidis and C. Stephanidis, "Unified user interface design: designing universally accessible interactions," *Interacting with computers*, vol. 16, no. 2, pp. 243–270, 2004.

[62] C.-M. Chang, *Influences of new media communication on the deaf/hard of hearing as reflected in interaction design*. PhD thesis, Nottingham Trent University, 2016.

[63] J. Korte, L. E. Potter, and S. Nielsen, "An experience in requirements prototyping with young deaf children," *Journal of Usability Studies*, vol. 10, no. 4, pp. 195–214, 2015.

[64] B. N. Shiver and R. J. Wolfe, "Evaluating alternatives for better deaf accessibility to selected web-based multimedia," in *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, pp. 231–238, ACM, 2015.

[65] J. S. Dumas and J. Redish, *A practical guide to usability testing*. Intellect Books, 1999.

[66] J. Rubin and D. Chisnell, *Handbook of usability testing: how to plan, design and conduct effective tests*. John Wiley & Sons, New York, 2008.

[67] M. R. Ebling and B. E. John, "On the contributions of different empirical data in usability testing," in *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, pp. 289–296, ACM, 2000.

[68] L. L. Downey, "Group usability testing: Evolution in usability techniques," *Journal of Usability Studies*, vol. 2, no. 3, pp. 133–144, 2007.

[69] S. Riihiaho, *Experiences with usability testing: Effects of thinking aloud and moderator presence*. PhD thesis, Aalto University, 2015.

[70] E. De Kock, J. Van Biljon, and M. Pretorius, "Usability evaluation methods: Mind the gaps," in *Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, pp. 122–131, ACM, 2009.

[71] A. Kaikkonen, A. Kekäläinen, M. Cankar, T. Kallio, and A. Kankainen, "Usability testing of mobile applications: A comparison between laboratory and field testing," *Journal of Usability studies*, vol. 1, no. 1, pp. 4–16, 2005.

[72] I. Stewart and W. Mckee, "The application of voice recognition technology to the development and presentation of complex engineering terminology to hearing impaired students," *The Institution of Electrical Engineers. lEE, Savoy Place, London, UK*, 2003.

[73] W. X. Fen and X. J. Cheng, "Using speech recognition technology to support education for deaf students," in *2010 2nd IEEE International Conference on Information Management and Engineering*, 2010.

[74] M. Jun and X. J. Cheng, "The exploration of the strategies and skills of effective use of voice recognition software in the classroom for deaf students," in *Future Networks, 2010. ICFN'10. Second International Conference on*, pp. 420–423, IEEE, 2010.

[75] J. Jiménez, A. M. Iglesias, J. F. López, J. Hernández, and B. Ruiz, "Tablet pc and head mounted display for live closed captioning in education," in *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, pp. 885–886, IEEE, 2011.

[76] R. Ranchal, T. Taber-Doughty, Y. Guo, K. Bain, H. Martin, J. P. Robinson, and B. S. Duerstock, "Using speech recognition for real-time captioning and lecture transcription in the classroom," *IEEE Transactions on Learning Technologies*, vol. 6, no. 4, pp. 299–311, 2013.

[77] S. Kawas, G. Karalis, T. Wen, and R. E. Ladner, "Improving real-time captioning experiences for deaf and hard of hearing students," in *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 15–23, ACM, 2016.

[78] T. Matthews, S. Carter, C. Pai, J. Fong, and J. Mankoff, "Scribe4me: Evaluating a mobile sound transcription tool for the deaf," in *International Conference on Ubiquitous Computing*, pp. 159–176, Springer, 2006.

[79] J. Van Gelder, I. Van Peer, and D. Aliakseyeu, "Transcription table: text support during meetings," in *IFIP Conference on Human-Computer Interaction*, pp. 1002–1005, Springer, 2005.

[80] S. Lee, S. Kang, H. Ko, J. Yoon, and M. Keum, "Dialogue enabling speech-to-text user assistive agent with auditory perceptual beamforming for hearing-impaired," in *2013 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 360–361, IEEE, 2013.

[81] R. S. Kushalnagar, G. W. Behm, A. W. Kelstone, and S. Ali, "Tracked speech-to-text display: Enhancing accessibility and readability of real-time speech-to-text," in *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, pp. 223–230, ACM, 2015.

[82] D. Bragg, N. Huynh, and R. E. Ladner, "A personalizable mobile sound detector app design for deaf and hard-of-hearing users," in *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 3–13, ACM, 2016.

[83] M. Mielke and R. Brück, "A pilot study about the smartwatch as assistive device for deaf people," in *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, pp. 301–302, ACM, 2015.

[84] M. J. Hirayama, "A communication aid for hearing impaired persons using mobile smart phones," in *International Conference on Mobile IT Convergence*, pp. 58–63, IEEE, 2011.

[85] S. S. Prietch, N. S. de Souza, and L. V. L. Filgueiras, "Application requirements for deaf students to use in inclusive classrooms," in *Proceedings of the Latin American Conference on Human Computer Interaction*, p. 5, ACM, 2015.

[86] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584, IEEE Signal Processing Society, December 2011.

[87] T. Alumäe and K. Kaljurand, "Open and extendable speech recognition application architecture for mobile environments," in *Spoken Language Technologies for Under-Resourced Languages*, 2012.

[88] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of dnns with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[89] D. Iskra, B. Grosskopf, K. Marasek, H. Heuvel, F. Diehl, and A. Kiessling, "SPEECON – speech databases for consumer devices: Database specification and validation," *ELRA, European Language Resources Association, Paris*, 2002.

[90] Y.-Y. Wang, A. Acero, and C. Chelba, "Is word error rate a good indicator for spoken language understanding accuracy," in *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*, pp. 577–582, IEEE, 2003.

[91] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2176–2184, 2016.

[92] ITU-R BS.1116-1, "Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems," recommendation, International Telecommunication Union, Geneva, October 1997.

[93] A. Järvinen, "Kuunteluhuoneen suunnittelu ja mallinnus," Master's thesis, Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, 1999.

[94] R. K. Furness, "Ambisonics – an overview," in *Audio Engineering Society Conference: 8th International Conference: The Sound of Audio*, May 1990.

[95] V. Pulkki, "Directional audio coding in spatial sound reproduction and stereo upmixing," in *Audio Engineering Society Conference: 28th International Conference: The Future of Audio Technology – Surround and Beyond*, (Piteå, Sweden), June 2006.

[96] MathWorks MATLAB documentation, “Box plots.” Online, accessed 20.9.2017. Available at https://se.mathworks.com/help/stats/box-plots.html.

A Prototype Source Code

The source code of the Conversation Assistant prototype implemented with the Python programming language (version 3).

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
# Copyright (c) 2017 Juri Lukkarila
# Copyright (c) 2013 Tanel Alumae
# Copyright (c) 2008 Carnegie Mellon University.
#
# Inspired by the CMU Sphinx's Pocketsphinx Gstreamer plugin demo (which has BSD license)
#
# Licence: BSD

import sys
import os
import gi
gi.require_version('Gst', '1.0')
gi.require_version('Gtk', '3.0')
from gi.repository import GObject, Gst, Gtk, Gdk, Pango
GObject.threads_init()
Gdk.threads_init()

Gst.init(None)

class DemoApp(object):

    def __init__(self):

        self.init_gui()
        self.init_gst()

    def init_gui(self):

        self.window = Gtk.Window()
        self.window.connect("destroy", self.quit)
        self.window.set_default_size(800, 1000)
        self.window.set_border_width(10)
        self.window.set_title("KESKUSTELUAVUSTIN")

        # layout
        vbox = Gtk.VBox()

        # partial result text view
        self.text_partial = Gtk.TextView()
        self.text_partial.override_font(Pango.FontDescription("DejaVu Sans Mono 20"))
        self.textbuf_partial = self.text_partial.get_buffer()
        self.text_partial.set_wrap_mode(Gtk.WrapMode.WORD)
        self.text_partial.set_cursor_visible(False)

        # scrolling window for final result
        self.scrolled = Gtk.ScrolledWindow()

        # final result text view
        self.text_final = Gtk.TextView()
        self.text_final.override_font(Pango.FontDescription("DejaVu Sans Mono 20"))
        self.textbuf_final = self.text_final.get_buffer()
        self.text_final.set_wrap_mode(Gtk.WrapMode.WORD)
        self.text_final.set_cursor_visible(False)

        self.scrolled.add(self.text_final)

        # button
        self.button = Gtk.Button("Puhu")
        self.button.connect('clicked', self.button_clicked)
```

```
        vbox.pack_start(self.scrolled, True, True, 4)
        vbox.pack_start(self.text_partial, False, False, 1)
        vbox.pack_start(self.button, False, False, 8)

        self.window.add(vbox)
        self.window.show_all()

    def quit(self, window):
        Gtk.main_quit()

    def init_gst(self):

        self.pulsesrc = Gst.ElementFactory.make("pulsesrc", "pulsesrc")

        if self.pulsesrc == None:
            print >> sys.stderr, "Error loading pulsesrc GST plugin. You need the gstreamer1.0-pulseaudio package"
            sys.exit()

        # TODO: get audio level and ignore input below threshold!
        self.level = Gst.ElementFactory.make("level", "level")

        self.audioconvert  = Gst.ElementFactory.make("audioconvert", "audioconvert")
        self.audioresample = Gst.ElementFactory.make("audioresample", "audioresample")
        self.asr           = Gst.ElementFactory.make("kaldinnet2onlinedecoder", "asr")
        self.fakesink      = Gst.ElementFactory.make("fakesink", "fakesink")

        if self.asr:
            model_file = "final.mdl"
            if not os.path.isfile(model_file):
                print >> sys.stderr, "Model Not Found"
                sys.exit(1)
            self.asr.set_property("use-threaded-decoder", True)
            self.asr.set_property("acoustic-scale", 0.09)
            self.asr.set_property("beam", 10.0)
            self.asr.set_property("lattice-beam", 6.0)
            self.asr.set_property("max-active", 10000)
            self.asr.set_property("model", "final.mdl")
            self.asr.set_property("fst", "graph_finnish_morph_5gram/HCLG.fst")
            self.asr.set_property("word-syms", "graph_finnish_morph_5gram/words.txt")
            self.asr.set_property("feature-type", "mfcc")
            self.asr.set_property("mfcc-config", "conf/mfcc.conf")
            self.asr.set_property("ivector-extraction-config", "conf/ivector_extractor.conf")
            self.asr.set_property("do-endpointing", True)
            self.asr.set_property("endpoint-silence-phones", "1:2:3:4:5")
            self.asr.set_property("chunk-length-in-secs", 0.2)

        else:
            print >> sys.stderr, "Couldn't create the kaldinnet2onlinedecoder element. "
            if os.environ.has_key("GST_PLUGIN_PATH"):
                print >> sys.stderr, "Have you compiled the Kaldi GStreamer plugin?"
            else:
                print >> sys.stderr, "You probably need to set the GST_PLUGIN_PATH envoronment variable"
                print >> sys.stderr, "Try running: GST_PLUGIN_PATH=../src %s" % sys.argv[0]
            sys.exit()

        # initially silence the decoder
        self.asr.set_property("silent", True)

        self.pipeline = Gst.Pipeline()
        for element in [self.pulsesrc, self.audioconvert, self.audioresample, self.asr, self.fakesink]:
            self.pipeline.add(element)
        self.pulsesrc.link(self.audioconvert)
        self.audioconvert.link(self.audioresample)
        self.audioresample.link(self.asr)
        self.asr.link(self.fakesink)

        self.asr.connect('partial-result', self._on_partial_result)
```

```
        self.asr.connect('final-result', self._on_final_result)
        self.pipeline.set_state(Gst.State.PLAYING)

    def _on_partial_result(self, asr, hyp):
        """Delete any previous selection, insert text and select it."""
        Gdk.threads_enter()
        # All this stuff appears as one single action
        self.textbuf_partial.begin_user_action()

        self.textbuf_partial.delete_selection(True, self.text_partial.get_editable())

        new_text = hyp.replace("+ +", "").replace("+", "")
        if (len(new_text) > 1):

            #print(new_text)
            if (hyp[0] is "+"):
                    # go back one char
                ins = self.textbuf_partial.get_cursor()
                iter = self.textbuf_partial.get_iter_at_mark(ins)
                iter.backward_chars(1)
                self.textbuf_partial.move_mark(ins, iter)

            self.textbuf_partial.insert_at_cursor(new_text)
            ins = self.textbuf_partial.get_insert()
            iter = self.textbuf_partial.get_iter_at_mark(ins)
            iter.backward_chars(len(new_text))
            self.textbuf_partial.move_mark(ins, iter)

        self.textbuf_partial.end_user_action()
        Gdk.threads_leave()

    def _on_final_result(self, asr, hyp):
        Gdk.threads_enter()
        """Insert the final result."""
        # All this stuff appears as one single action
        self.textbuf_final.begin_user_action()
        self.textbuf_partial.begin_user_action()
        if (len(hyp) > 1):
            self.textbuf_partial.delete(self.textbuf_partial.get_start_iter(), self.textbuf_partial.get_end_iter())
            new_text = hyp.replace("+ +", "").replace("+","")
            self.textbuf_final.insert_at_cursor(new_text)
            self.textbuf_final.insert_at_cursor("\n\n")

            self.text_final.scroll_mark_onscreen(self.textbuf_final.get_insert())

        self.textbuf_partial.end_user_action()
        self.textbuf_final.end_user_action()
        Gdk.threads_leave()

    def button_clicked(self, button):

        if button.get_label() == "Puhu":
            button.set_label("Pysäytä")
            self.asr.set_property("silent", False)
        else:
            button.set_label("Puhu")
            self.asr.set_property("silent", True)

if __name__ == '__main__':
    app = DemoApp()
    Gdk.threads_enter()
    Gtk.main()
    Gdk.threads_leave()
```

B Questionnaire

Printed pages of the questionnaire from the Google Forms web application used in the user testing.

KESKUSTELUAVUSTIN

Koehenkilön taustatiedot. Kaikkiin kysymyksiin ei ole pakko vastata. Taustatietoja käytetään ainoastaan tulosten arvioimisessa ja tilastoinnissa.

*Required

1. Nimi *

Your answer

2. Sähköposti *

Your answer

3. Ikä

Your answer

4. Elämäntilanne

- ◯ Opiskelija
- ◯ Työelämässä
- ◯ Työtön
- ◯ Eläkeläinen

5. Omistatko älypuhelimen?

Your answer

6. Omistatko tabletin?

Your answer

7. Käytätkö jotain apuvälinettä kuulemiseen? Jos kyllä niin mitä?

Your answer

8. Oletko käyttänyt aikaisemmin jotain puheentunnistukseen perustuvaa sovellusta tai palvelua?

- ◯ Kyllä
- ◯ En

9. Jos vastasit kyllä kysymykseen 8, kerrotko tarkemmin mitä sovellusta. Voit myös lyhyesti kuvailla kokemusta: mitä hyvää, mitä parannettavaa, kuinka usein käytät sovellusta ja missä tilanteissa?

Your answer

10. Kuinka taitava olet mielestäsi käyttämään tietotekniikkaa arkielämässä? (mobiililaitteet, tietokone)

	1	2	3	4	5	6	7	
Huono	◯	◯	◯	◯	◯	◯	◯	Hyvä

NEXT

Page 1 of 4

Never submit passwords through Google Forms.

KESKUSTELUAVUSTIN

*Required

Osio 1: Sananselitys

Arvioi Keskusteluavustinta esitettyjen kysymyksien perusteella asteikolla 1-7. Voit myös lisätä halutessasi sanallisia kommentteja jokaisen kysymyksen tekstikenttään.

11. Auttoiko Keskusteluavustin ymmärtämään puhetta? *

	1	2	3	4	5	6	7	
Ei ollenkaan	○	○	○	○	○	○	○	Todella paljon

11. lisää halutessasi kommentteja tähän

Your answer

12. Oliko Keskusteluavustimen käyttäminen sujuvaa? *

	1	2	3	4	5	6	7	
Ei sujuvaa	○	○	○	○	○	○	○	Todella sujuvaa

12. lisää halutessasi kommentteja tähän

Your answer

13. Haittasiko Keskusteluavustin puhujan seuraamista? *

	1	2	3	4	5	6	7	
Ei ollenkaan	○	○	○	○	○	○	○	Haittasi paljon

13. lisää halutessasi kommentteja tähän

Your answer

14. Toimiko puheentunnistus tarpeeksi nopeasti? *

	1	2	3	4	5	6	7	
Liian hidas	○	○	○	○	○	○	○	Tarpeeksi nopea

14. lisää halutessasi kommentteja tähän

Your answer

15. Tunnistiko Keskusteluavustin puheen tarpeeksi hyvin? (puhe tunnistui oikein) *

	1	2	3	4	5	6	7	
Käyttökelvoton	○	○	○	○	○	○	○	Tarpeeksi hyvä

15. lisää halutessasi kommentteja tähän

Your answer

BACK NEXT Page 2 of 4

Never submit passwords through Google Forms.

KESKUSTELUAVUSTIN

*Required

Osio 2: Vapaa keskustelu

Arvioi Keskusteluavustinta esitettyjen kysymyksien perusteella asteikolla 1-7. Voit myös lisätä halutessasi sanallisia kommentteja jokaisen kysymyksen tekstikenttään.

16. Auttoiko Keskusteluavustin ymmärtämään puhetta? *

	1	2	3	4	5	6	7	
Ei ollenkaan	○	○	○	○	○	○	○	Todella paljon

16. lisää halutessasi kommentteja tähän

Your answer

17. Oliko Keskusteluavustimen käyttäminen sujuvaa? *

	1	2	3	4	5	6	7	
Ei sujuvaa	○	○	○	○	○	○	○	Todella sujuvaa

17. lisää halutessasi kommentteja tähän

Your answer

18. Haittasiko Keskusteluavustin puhujan seuraamista? *

	1	2	3	4	5	6	7	
Ei ollenkaan	○	○	○	○	○	○	○	Paljon

18. lisää halutessasi kommentteja tähän

Your answer

19. Hidastiko Keskusteluavustin keskustelua? *

	1	2	3	4	5	6	7	
Ei ollenkaan	○	○	○	○	○	○	○	Paljon

19. lisää halutessasi kommentteja tähän

Your answer

20. Toimiko puheentunnistus tarpeeksi nopeasti? *

	1	2	3	4	5	6	7	
Liian hidas	○	○	○	○	○	○	○	Tarpeeksi nopea

20. lisää halutessasi kommentteja tähän

Your answer

21. Tunnistiko Keskusteluavustin puheen tarpeeksi hyvin? *

	1	2	3	4	5	6	7	
Käyttökelvoton	○	○	○	○	○	○	○	Tarpeeksi hyvä

21. lisää halutessasi kommentteja tähän

Your answer

BACK NEXT Page 3 of 4

Never submit passwords through Google Forms.

KESKUSTELUAVUSTIN

*Required

Loppukysely

Vastaa esitettyihin kysymyksiin koko koetilaisuuden kokemuksien perusteella. Voit jättää vastauksen tyhjäksi jos sinulla ei ole mitään kommentoitavaa johonkin sanalliseen kysymykseen.

22. Oliko Keskusteluavustimesta hyötyä koetilanteissa? *

	1	2	3	4	5	6	7	
Ei ollenkaan	○	○	○	○	○	○	○	Paljon

23. Selitätkö vastauksesi edelliseen kysymykseen (22.) *

Your answer

24. Pidätkö tärkeänä sitä, että Keskusteluavustimen fontin kokoa ja väriä voisi säätää vapaasti? *

	1	2	3	4	5	6	7	
Ei ollenkaan tärkeää	○	○	○	○	○	○	○	Todella tärkeää

24. lisää halutessasi kommentteja tähän

Your answer

25. Mikä Keskusteluavustimessa oli hyvää?

Your answer

26. Mitä Keskusteluavustimessa pitäisi kehittää?

Your answer

27. Mitä ominaisuuksia toivoisit löytyvän Keskusteluavustimen tapaisesta sovelluksesta?

Your answer

28. Käyttäisitkö tai oletko jo käyttänyt Keskusteluavustimen tapaista sovellusta?

Your answer

29. Missä tilanteissa käyttäisit Keskusteluavustinta?

Your answer

30. Kuinka paljon olisit valmis maksamaan Keskusteluavustimen tapaisesta sovelluksesta kuukaudessa?

- ◯ 0€
- ◯ 1-5€
- ◯ 5-10€
- ◯ 10-20€
- ◯ 20-30€
- ◯ Enemmän kuin 30€

30. lisää halutessasi kommentteja tähän

Your answer

31. Kuinka paljon olisit valmis maksamaan Keskusteluavustimen tapaisesta sovelluksesta kertamaksuna?

- ◯ 0€
- ◯ 1-5€
- ◯ 5-10€
- ◯ 10-20€
- ◯ 20-30€
- ◯ Enemmän kuin 30€

31. lisää halutessasi kommentteja tähän

Your answer

32. Maksaisitko Keskusteluavustimen tapaisesta sovelluksesta mieluummin kertamaksun vai kuukausimaksua?

- ◯ Kertamaksu
- ◯ Kuukausimaksu

32. lisää halutessasi kommentteja tähän

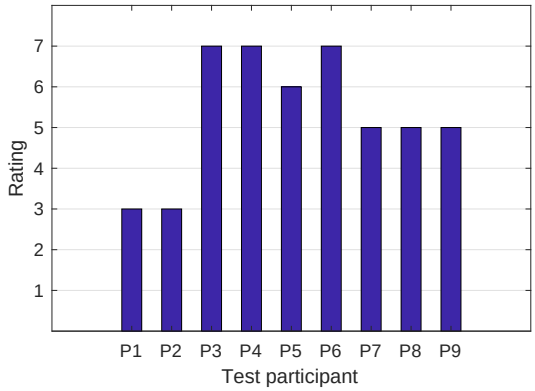
Your answer

BACK SUBMIT Page 4 of 4

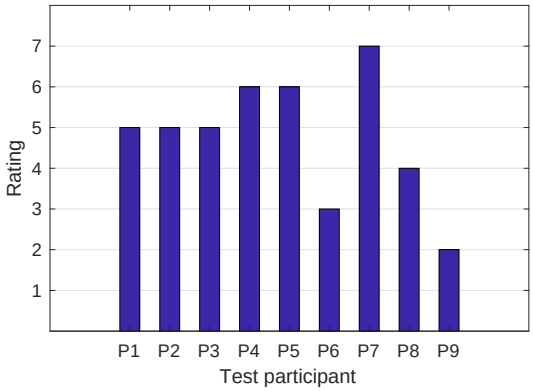
Never submit passwords through Google Forms.

C Questionnaire Answers

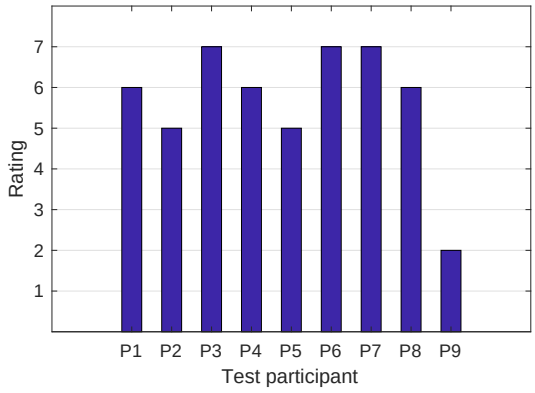
The raw data from the user tests is included here. Each set of answers is identified by the number of the question, which corresponds to the number reported at the questionnaire listing in section 4.4. On the x-axis, P1 to P9 refer to the test participants. Values on the y-axis are the numerical answers on the scale from 1 to 7.



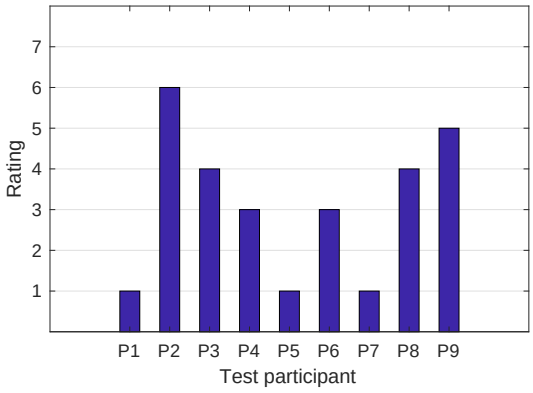
Q10



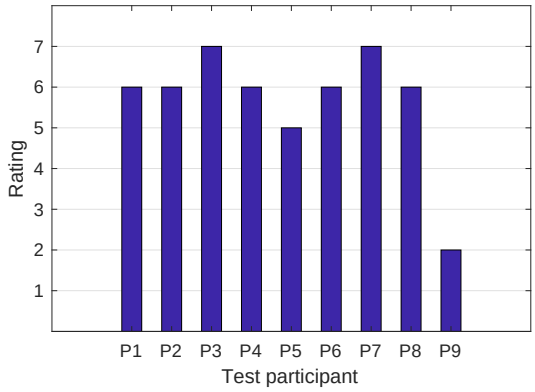
Q11



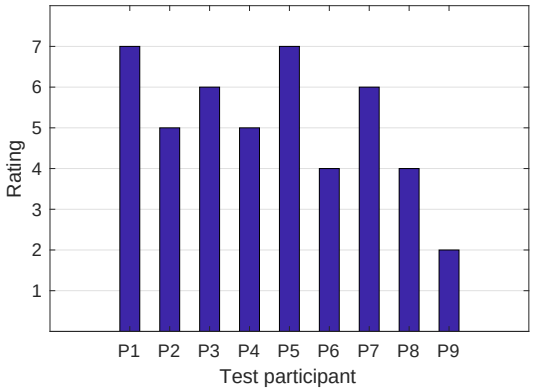
Q12



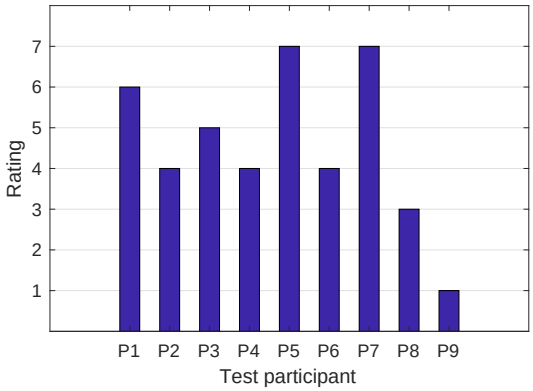
Q13



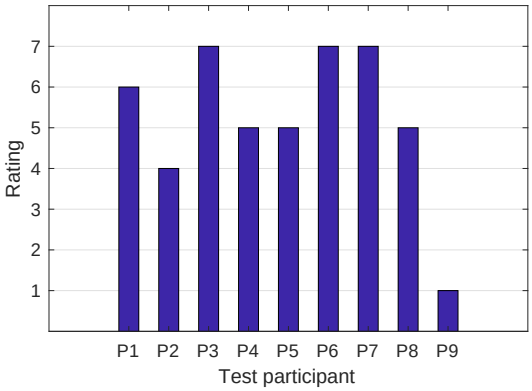
Q14



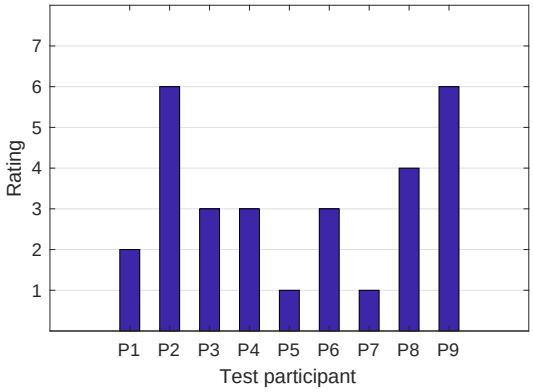
Q15



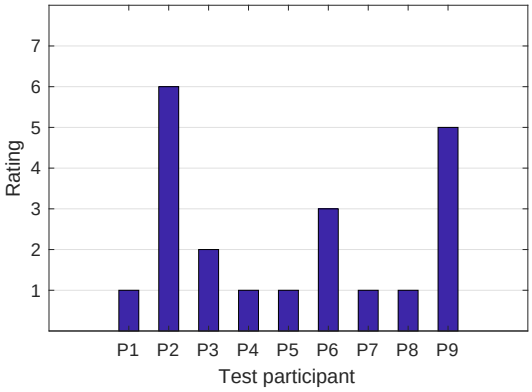
Q16



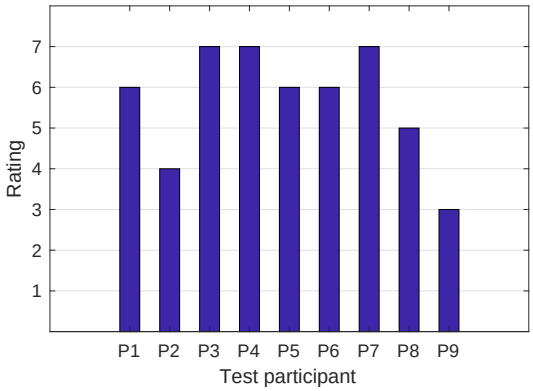
Q17



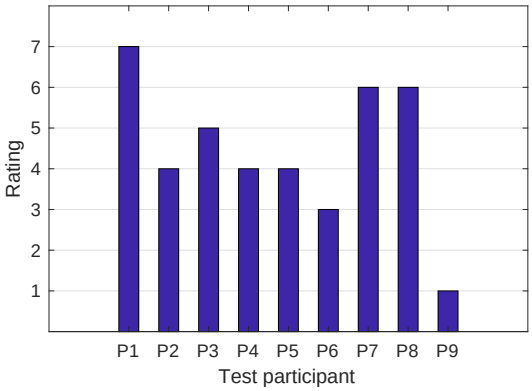
Q18



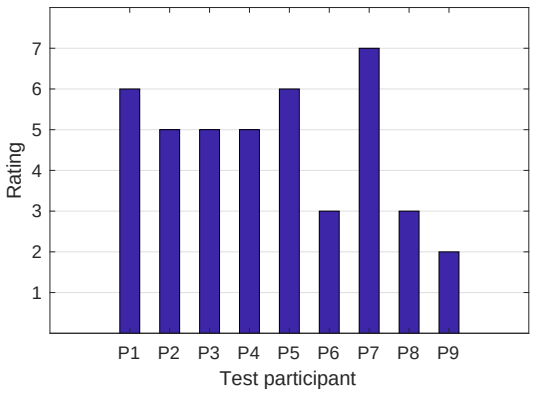
Q19



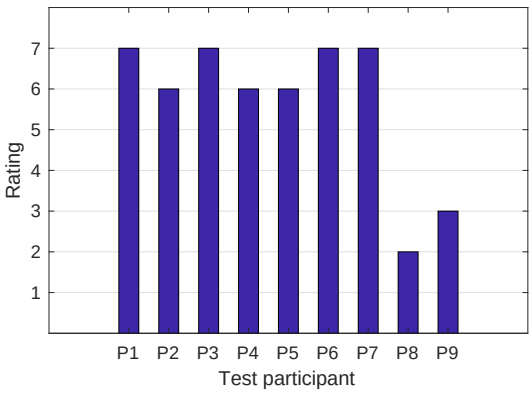
Q20



Q21



Q22



Q23