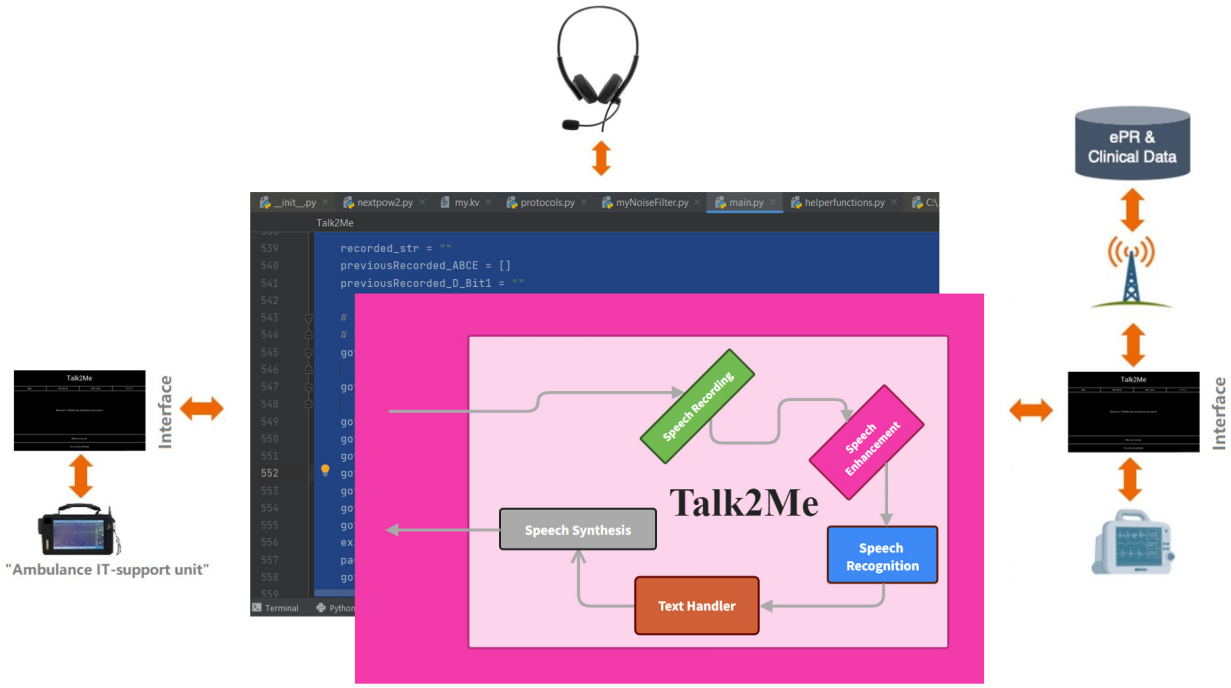
CHALMERS
UNIVERSITY OF TECHNOLOGY



Talk2Me

A Voice Controlled User Interface Used In The Initial Ambulance Care Process

Master's thesis in Electrical Engineering

Jane Jardebrand

Master's Thesis in The Master Degree Program, Electrical Engineering

Chalmers University of Technology
Gothenburg, Sweden 2022
www.chalmers.se

Master's thesis 2021

Talk2Me

A Voice Controlled User Interface Used In The Initial Ambulance Care Process

JANE JARDEBRAND

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2021

A Voice Controlled User Interface Used In The Initial Ambulance Care Process

JANE JARDEBRAND

Main supervisor: Stefan Candefjord
Associate Professor, Department of Electrical
Engineering, Chalmers University of Technology

Co-supervisor: Anna Bakidou
PhD Student, Department of Electrical Engineering
Chalmers University of Technology and University of Borås

Examiner: Bengt Arne Sjöqvist
Professor of Practice emeritus, Department of Electrical Engineering
Chalmers University of Technologyt

Master's Thesis 2021
Department of Electrical Engineering / Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A Voice Controlled User Interface Used In The Initial Ambulance Care Process designed by Python and Kivy.

Talk2Me
A Voice Controlled User Interface Used In The Initial Ambulance Care Process
Jane Jardebrand
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Prehospital care is medical care provided outside the hospital setting by various organizations but in most cases an ambulance service. Prehospital care is time-sensitive. To improve the medical outcome for the patient, ambulance personnel are expected to provide accurate information about the patient and effective medical care. In a stressful and problematic situation, it is unlikely that this information can be obtained as quickly as needed from the ambulance personnel using standard tools like tapping in information on a computer and communicating guidelines, results, etc. to the ambulance personnel using a computer screen. To deal with this voice recognition and speech synthesis opening new opportunities, together with real-time AI-based clinical decision support give a possible solution.

The goal of this master's thesis is to test whether voice control is suitable for the real-time registration of information obtained during a patient's assessment. A prototype of a demonstrator is designed that deals with issues related to applying voice recognition in an ambulance setting and how to deal with user feedback through a number of tests on a limited number of individuals. The prototype is a voice-controlled user interface, a platform that supports the conversation between humans and machines. It uses the technology of Speech Recognition and Speech Synthesis to make the conversation possible. A Spectrum-Subtraction-Noise filter is designed for speech enhancement. As a test case the first patient assessment, the primary survey, referred to as ABCDE has been chosen. A series of tests are implemented for prototype evaluation that focuses on recognition accuracy and speed. For accuracy, the algorithm guarantees 100% for input accuracy and more than 70% for speech recognition accuracy. The speed to recognize an oral-input command is about 1.8s.

To be used in ambulances requires 100% accuracy and efficient response times. The result from this thesis is basically satisfied the requirement. It shows that speech is a suitable method for information handling in real-time. This thesis supplies a successful prototype for future work.

Keywords: Prehospital Care, ABCDE, Kivy, Speech Recognition, Speech Synthesis, Speech Enhancement, Spectrum Subtraction Noise Filter

Acknowledgements

Large gratitude is directed to Bengt-Arne, Stefan, and Anna. Thanks for the time we shared together in those months. All of you have spent your treasured time on me. Thanks for your help and instructions! I especially want to thank Anna, you always helped and supported me in every way you could.

Thanks to Arto Kemppainen and Marcus Martinsson! It is your fantastic Pilot job to give me a good start. Your good idea makes me, a new Python starter, easily come into the master thesis project. I used your code as the basis for the thesis.

And big thanks to the 8 testers. Your participation has provided valuable data for this thesis.

Jane Jardebrand, Gothenburg, January 2022

Abbreviations

PHTLS PreHospital Life Support
AMLS Advanced Medical Life Support
ABCDE Airway, Breathing, Circulation, Disability, Exposure
GUI Graphical User Interface
ICT Information and Communications Technologies
ANN Artificial Neural Network
NLP Natural Language Processing
SNR Signal-to-Noise Ratio
MFCC Mel Frequency Cepstral Coefficents
WGN White Gaussian Noise
ReLu Rectified Linear Unit

Contents

List of Figures

List of Tables

1

Introduction

When an accident occurs or when an illness strikes unexpectedly, prehospital care is the initial health care people get [1]. Prehospital care includes not only the transport but also assessments and medical care [2]. Assessments as a structured workflow in prehospital care vary for different patients [3]. As test case the first patient assessment, the primary survey, Airway, Breathing, Circulation, Disability, Exposure (ABCDE) Assessment is one of the most acceptable assessments to investigate the patient's vital status, to identify essential maybe life-threatening obstacles, and address these in the first treatments and procedures carried out [4]. The protocol states how these vitals shall be checked, and their status documented. This survey is well-suited as a test case since it is short and highly standardized regarding the procedure as well as input data alternatives for each step. The traditional method for assessment documentation is to make the assessment on the patient's status first, then make the document manually on paper, electronic devices, or sometimes a combination of both [5]. The traditional data recording method works well in many cases, but when the situation becomes more stressful, and maybe also in the other situations which need quick documentation, new technology is expected. Prompt prehospital care has a major influence on patients' late prognosis [6] and the speed of response of the prehospital mergency medical services is critical to the survival of patients [7]. The ambulance nurses need to complete assessments and documentation in the shortest possible time for the patient's immediate care. Digital Health support of real-time documentation gives a solution.

In this thesis, a voice-controlled interface is designed for real-time documentation of ABCDE assessment. Real-time voice functionality requires AI technology to provide support, such as speech recognition and speech synthesis. There is two possible selection for speech recognition, a self-built Artificial Neural Network (ANN) and the commercial tools in the market. For the self-built ANN, it is fast, robust, but it needs enormous training samples that are impossible to obtain in a short time. ANN is good, but unfortunately cannot be used in this paper. Instead, this thesis uses the Python library "SpeechRecognition" for speech recognition. The commercial tool is used for synthesis too. To improve the effect of recognition, speech enhancement is designed with a Spectrum-Subtraction-Noise.

To test the performance of the prototype, two groups of tests: Self-tests and Public-tests are designed to evaluate if speech input is a suitable method for real-time documentation. Self-tests were completed by the thesis writer. A total of eight testers took part in the Public-Tests.

Test results show that as a voice-controlled user interface, this thesis contributes a good algorithm of the prototype building. Speech is a suitable method for information handling in real-time.

1.1 Prehospital Care

Prehospital care is medical care provided outside the hospital setting by various organizations. In most cases, it is the ambulance [8]. In reality, in the event of an accident or some sudden illness, it is common for people to first use their own judgment to ascertain whether they need external professional emergency agencies to obtain medical help. For minor traumas and less serious illnesses, people use self-medication to solve the problem. For serious injuries and illnesses, people usually call the emergency services through medical alerts in the hope of receiving guidance and assistance. The emergency medical facilities make the incident/illness assessment and prioritization on the reported cases. The calls for the ambulance service exhibit a large variation when it comes to diseases and type of situations to handle as well as its emergency and prioritization. In those situations where medical assessment is needed these often follow predefined standardized protocols like PreHospital Life Support (PHTLS) and Advanced Medical Life Support (AMLS). In addition, there are also a set of other standardized protocols and procedures like triaging, care routines, and pathways that governs the care and decisions being made, but also logistics i.e., where the patient shall be transported to obtain optimal care. During transport, patients receive not only the transportation service but also the medical care provided by the ambulance nurses [9]. Popular prehospital care includes a quick assessment and evaluation of the patient's condition and supplies the necessary information for the next measures and medical treatment [1]. It is the ambulance nurses who make patient assessments, prioritization and supply the care.

Prehospital care includes multiple steps from the occurrence of illness or injury to the healthcare provider that are demonstrated in the prehospital care chain, see Figure 1.1.

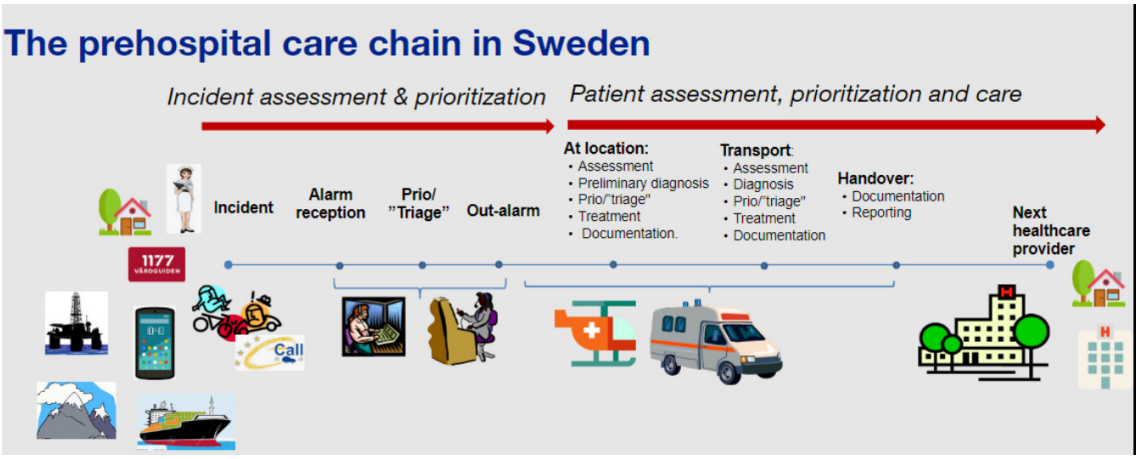


Figure 1.1: The prehospital care chain in Sweden [10]

In recent decades, prehospital care has changed dramatically to adapt to the requirement of society. More advanced assessments and decisions for the patients are taken in the prehospital care system [11]. Specific to prehospital care performance and Information and Communications Technologies (ICT) is vital [7]. Timely and accurate information is essential for decision-making [12]. Prehospital care supplier has the need for real-time data to generate useful information and help them provide optimal care [12]. Supervi-

sors also need real-time data so they are continuously and appropriately informed about site performance [12]. Real-time documentation about the patient assessment systems is concerned in this thesis.

1.2 ABCDE Assessment

The ambulance nurses make an assessment of the patient [13]. The underlying principle is to assess and treat the patient and make the initial assessment completely [13]. When it is necessary, the ambulance nurses need to re-assessment regularly and evaluate the effects of treatment. The basics of the ABCDE approach are shown in Table 1.1:

Letter		Life-threatening condition
A	Airway	Airway blockage, cervical spine injury
B	Breathing	Tension pneumothorax, pulmonary oedema, bronchospasm
C	Circulation	Shock(Hypovolaemic, obstructive, distributive, cardiogenic)
D	Disability	Seizure, hypoglycaemia, meningitis, intracranial haemorrhage or infarction
E	Exposure	Hypothermia or hyperthermia, critical skin conditions such as fasciitis or urticaria

Table 1.1: The basics of the ABCDE Assessment [14]

Airway obstruction is a medical emergency and the patient is in a dangerous situation that can cause hypoxia, risk damage to the brain, kidneys, and heart [15]. If serious, it can lead to death [15]. The nurse needs to find the reason that causes obstruction and assess the degree of airway obstruction by the patient's symptoms to identify if it is a partial airway obstruction or complete obstruction [15]. In most cases, only simple methods of airway clearance are required. For the serious, tracheal intubation and high oxygen concentration may be required [15]. If necessary, it is needed to obtain expert help immediately [15]. For Breathing assessment, it is important to diagnose and treat immediately life-threatening conditions [15]. The nurse needs to look, listen and feel to know if the patient is breathing [16]. Breathing speed is expected to be assessed and abnormal breath sounds need to be detected for supplying different care management [16]. In the Circulation assessment, the nurse needs to detect and determine the patient's circulatory status through a series of observations and measurements such as internal and external signs of bleeding, and check for pericardial tamponade and blood pressure [15][16]. Disability Assessment includes assessment of the level of consciousness. The nurse needs to check the movement and sensation in all four limbs and find if there exists abnormal repetitive movements or shaking [16]. Measurements such as low blood glucose and pupils are also required [16]. The unconscious patients should care in the lateral position if their airway is not protected [15]. The entire body needs to examine for hidden injuries, rashes, bites, or other lesions if necessary[16].

1.3 Aim

This master's thesis deals with the design, testing, and evaluation of software for the real-time information handling of speech recognition and synthesis. A prototype is designed to document the care process in a structured way and provide real-time information handling

support. The prototype is a voice-controlled PC interface for prehospital care that focuses on ABCDE Assessment. Then through a series of tests to evaluate if speech is a suitable method for information handling in real-time.

1.4 Related Work

A prototype is designed for real-time information handling. The user can perform patient registration and ABCDE assessment by speech: The functions that the prototype can achieve are as follows:

1. Realize man-machine logical conversation.
2. Finish ABCDE Approach through the conversation. The user decides which assessment should be made and decide when ABCDE is finished.
3. The user can go backward freely.
4. The user has permission to correct the old mistake.
5. Pause the system operation.
6. Exit the program.
7. Automated information collection after each ABCDE approach is completed.

Two groups of tests **Self-tests** and **Public-tests** are designed for the evaluation of the prototype.

1.5 Limitations

The purpose of the thesis is to evaluate if speech is a good method for information handling.

1. The end product of Talk2Me is a user interface expected to be used as a mobile app or an Andriod APK that should be installed in the headset with head-mounted display systems, such as the headset **Realwear HMT-1**. Because of the hardware constraint, the interface can only work in the windows operation system.
2. ABCDE Assessment is the only assessment implemented in this thesis. It locates in the part of **First Assessment** of the prototype.
3. To improve the performance of the prototype for speech recognition, a **Spectrum Reduction Filter** and a **Kalman Filter** are expected to be used for Speech Enhancement. Only the **Spectrum Reduction Filter** is finished. The function of the Kalman filter is finished but the operation speed is too low so only the filtering effect is shown but it is not actually used in this thesis.
4. A user interface is designed as the communication platform between the user and the machine. The user interface(**GUI**) is very simple. Python and Kivy are used for the implementation of GUI. In Kivy, the widgets as the base building block of GUI interfaces [17] supply multiple function construction tools. Many interesting and useful widgets can be used for the GUI design but the method of activating them is all touched-based. Because Talk2Me is a voice-controlled interface and finding the way to use them needs time that leads it is hard to use those widgets in this thesis. The selection of widgets consists of Label, BoyLayout, GridLayout, etc.
5. For the testing design to the prototype, the number of testers is limited to 10 persons. But in practice, only 8 testers took the tests because of the global epidemic of COVID 19.

6. No professional testers. The main reason is that the prototype is in a too early stage that can only work on the windows computer. The installation of the software is complex and it is hard for those who have no professional knowledge to install the prototype on their own computer. That leads the care personnel will have trouble providing meaningful feedback. Their time is limited and should therefore only be asked for when the prototype can be used as intended on a mobile application. Another reason is that the thesis is expected to end at the end of August. Most of the care personals are on invocation.
7. Talk2Me is expected to be used in the ambulance. For the test design, a good simulation place is in a running private car or a running bus. In this thesis, these two places are not selected because it is not good to make tests in a running car for security and it is not acceptable to make tests in a running bus to avoid infringement of public rights. Case 4 is done in Gråbo center, a public place. The place selection should be carefully considered to have the best effects of the tests and that has the minimal effect on public rights.

1.6 Thesis Outline

The **Theory** chapter presents information and theories related to this thesis, such as ABCDE assessment, Speech Recognition, Speech Synthesis, Spectrum Subtraction Noise Filter. The **Methods** chapter presents how to implement the prototype and the design of the tests. **Results & Discussion** chapter, presents the results of the tests of the selected 10 testers and partly debugging results. Based on the results, the data analysis is presented. The Results analysis reflects many problems of this thesis. Those problems and phenomena are discussed in the Discussion. In **Conclusion & Future Work** chapter, the final conclusion is given as a qualitative assessment of this thesis. In **Future work** chapter, it represents the recommended suggestions that are not finished in this thesis. In appendix, it includes a list of tables and graphs related to the thesis. The last part is the **references** about this thesis.

2

Theory

This chapter introduces related knowledge for the thesis. Speech recognition and speech synthesis, branches of Natural Language Processing (NLP), are used to implement the function for communication between the PC and the user. To improve the accuracy, the Spectrum Reduction noise filter is used for speech enhancement.

2.1 Natural Language Processing and Mixed-initiative interaction

Natural language, as the human language, is a medium for communications and transactions, and Natural Language technologies are the driving force behind various applications[18]. NLP is the way to use computers to analyze texts and gather the knowledge to understand and use language [19]. NLP technologies are widely used in knowledge acquisition, information retrieval, language translation, and decision-support systems [19].

Mixed-initiative interaction is a flexible interaction between human and machine systems where the user and the system act as equal participants in an activity that contribute to the best suited [20][21]. Conversation between two people is of mixed-initiative where the control over the conversation is transferred from one person to another [22].

2.2 Speech Enhancements

Talk2Me is designed to be used in an ambulance. Ambulance transports the patients to the hospital. The background noise such as the siren, the conversation between the care person and the patient, the sound of the motor, the sounds outside of the ambulance during transportation, has a great influence on speech recognition [23]. It is the most common factor that affects and degrades speech recording [23]. This thesis is therefore concerned with the issue of noise handling. For Speech Enhancement, the thesis considers two noise filters for noise handling. A Spectrum-Subtraction-Noise filter as one of the first algorithms proposed for noise reduction is used to reduce noise first [24]. While for a Spectrum-Subtraction-Noise filter, the drawback is the presence of processing distortions which are also called remnant noise [25]. So a Kalman filter is designed to optimize the filtering effect. Data after the Spectrum-Subtraction-Noise filter is then sent to Kalman Filter to be handled for the second time that creates a serial connection for the two noise filters.

2.2.1 Spectrum-Subtraction-Noise Filter

The Spectrum Subtraction Noise Filter uses the spectrum subtraction method for noise reduction [23]. It is assumed that the noise signal is a wide-band and stationary noise. For a speech signal y(t), it is a sum of true signal x(t) and the noise signal n(t).

$$y(t) = x(t) + n(t) \tag{2.1}$$

Using Fourier transform, y(t), x(t) and n(t) are converted from time domain to the frequency domain Y(jw), X(jw) and N(jw). The speech signal in the frequency domain is:

$$Y(jw) = X(jw) + N(jw) \tag{2.2}$$

The parameters of noise and the true speech are unknown but they can be estimated as $\hat{N}(jw)$ and $\hat{X}(jw)$. $\hat{N}(jw)$ is the expected noise spectrum:

$$\hat{N}(jw) = \mathbf{E}\Big[\mid N(jw) \mid \Big] \approx \mid \bar{N}(jw) \mid = \frac{1}{k} \sum_{j=0}^{K-1} \mid N_i(jw) \mid \tag{2.3}$$

Then (2.2) can be written as:

$$\hat{Y}(jw) = \hat{X}(jw) + \hat{N}(jw) \tag{2.4}$$

$$\hat{N}_k(jw) = \mid \tilde{N}_k(jw) \mid = \lambda_n \cdot \mid \tilde{K}_{k-1}(jw) \mid + (1 - \lambda_n) \cdot \mid (N_k(jw) \mid \tag{2.5}$$

Here $\tilde{N}_k(jw)$ is the smoothed noise estimate in i-th frame. The value of filtering coefficient λ is used to control the amount of noise subtracted from the noise signal. $\lambda = 1$ for full noise subtraction and a commonly used value is $0.5 \leq \lambda_n \leq 0.9$ [26][23].

The error is defined as:

$$\varepsilon = N(jw) - \mathbf{E}[N(jw) \mid] \approx \mid \bar{N}(jw) \mid - \mathbf{E}[\mid N(jw) \mid] \tag{2.6}$$

2.2.2 Kalman Filter

Kalman filter is an estimation algorithm [27]. In 1960, Rudolf E. Kalman (May 19, 1930 – July 2, 2016) published a paper about Kalman Filter [27]. It uses mathematical and statistical methods to produce estimates of hidden variables based on inaccurate and uncertain measurement data [27]. Based on the past estimations, the filter gives a new prediction of the future state. The application of the Kalman filter is numerous and has become popular in many fields. It can be used in tracking objects, noise filtering, economies, multicomputer vision applications, and speech enhancement [28] [29]. The most famous usage was in the Apollo Project in the 1960s, which demonstrated its powerful function for navigation, which is required to make an estimation for the trajectories of manned spacecraft going to the Moon and back [30]. For discrete-time Kalman filter, the working principle is shown in Figure 2.1

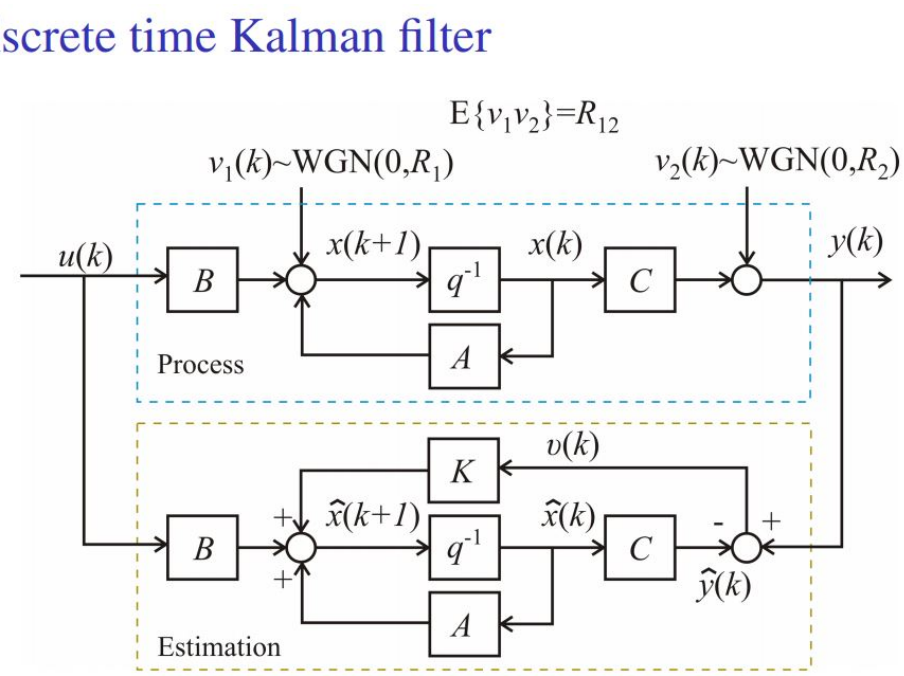


Figure 2.1: Discrete-time Kalman Filter [31]

In the Process block, the uncertain measurement data are generated as y. The Estimation block, as the predictor, generates the estimated data $\hat{y}$. A, B and C are the state matrix parameters, x is the state vector, u works as input/control vector and ν is the measurement noise vector.

For the predictor case, the discrete system is:

$$x(k+1) = Ax(k) + Bu(k) + N\nu_1(k) \tag{2.7}$$

$$y(k) = Cx(k) + Du(k) + \nu_2(k) \tag{2.8}$$

where ν_1 and ν_2 are White Gaussian Noise (WGN) as (2.9)

$$\begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} \sim WGN\left(0, \begin{bmatrix} R_1 & R_{12} \\ R_{12}^T & R2 \end{bmatrix}\right) \tag{2.9}$$

The estimating $\hat{x}(k+1) = A\bar{x}(k) + Bu(k) + \mathbf{K}(k)(y(k) + C\bar{x}(k) - Du(k))$.

The observer gain **K** and P as the variance of x, are calculated as bellow:

$$\mathbf{K}(k) = (AP(k)C^T + NR_{12})(CP(k)C^T + R_2)^{-1} \tag{2.10}$$

$$P(k+1) = AP(k)A^T + NR_1N^T - \mathbf{K}(AP(K)C^T + NR_{12})^T \tag{2.11}$$

2.2.3 Signal-to-Noise Ratio

Signal-to-Noise Ratio (SNR), used in science and engineering as a measurement to compare a desired signal and the background noise [32]. SNR is defined as the ratio of signal power to the noise power and is represented in decibels (dB) as follows [33]:

$$SNR = 10log(P_{signal}/P_{noise})(dB) \tag{2.12}$$
$$= 10log((A_{signal}/A_{noise})^2)(dB) \tag{2.13}$$
$$= 20log(A_{signal}/A_{noise})(dB) \tag{2.14}$$

Where P_{signal} and P_{noise} stands for the power of signal and noise and A_{signal} and A_{noise} stands for the amplitude of signal and noise.

2.3 Speech Recognition

Two ways to achieve speech recognition are discussed in this thesis. One is to use a commercial tool that is used in the prototype. The other is to establish an ANN that is only discussed in this thesis.

2.3.1 Commercial tools

In the market, there are many ready-made speech recognition libraries to realize speech recognition. The first top five software available are Kaldi, Julius, Fairseq, vosk, and Wav2Letter-Swedish [34]. Libraries CMU Sphinx, Google Speech Recognition, Google Cloud Speech API, Wit.ai, Microsoft Bing Voice Recognition are also popular in the market [35]. A part of them works offline, such as **CMU Sphinx** and **Snowboy Hotword Detection**. **Google Speech Recognition** works online.

2.3.2 Neural network

Another way for speech recognition is to establish an ANN as an artificial system that simulates humans' brains and has the ability to govern the optimal solution for many problems [36]. ANN mimics the human brain through artificial neurons. Neurons, also as a node, and the connection between neurons are the basis of a neural network [36] [36]. Neuron has an input and an output that can receive or send signals to communicate neurons and the environment. All neurons are connected to their neighbor neurons with connections via weights. The weight can modify the input or output value between neurons [36]. Weights, the basis of ANN's learning capability, are modified and determined during the training process [36]. The bias is an additional term that affects the output [37]. The structure of a single neuron is shown in Figure 2.2

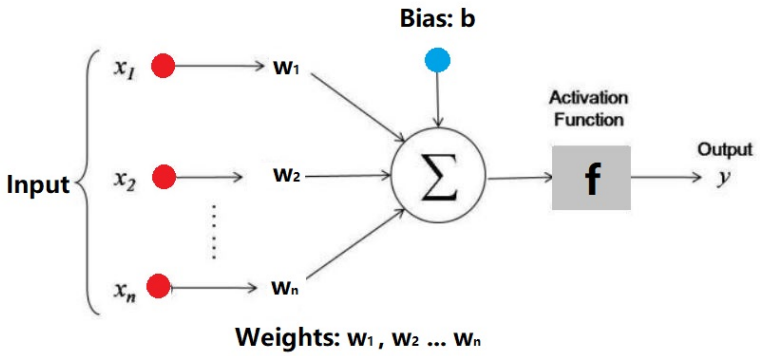


Figure 2.2: Structural diagram of a neuron [38]

From Figure 2.2, **y** is defined as follows:

$$y = \Sigma_{i=1}^{n} \omega_i x_i + b = \omega_1 x_1 + \omega_2 x_2 + ...\omega_n x_n + b \tag{2.15}$$

The values of weights determine the connections between neurons. The role of the bias is to shift the activation function to the left or right. Activation Function is a function that determines whether or not the activation of the neuron should take place by calculating weighted sum and bias [39] [40]. It is one of the important factors that affect the performance of the ANN [40]. The activation function is not a single definition but of different types. Activation functions are divided into two types, linear and non-linear activation functions. Non-linear activation functions, such as Sigmoid Function, Tanh Function, Rectified Linear Unit (ReLu), Leaky ReLu and, PReLu, RReLU, and ELU can be used for an ANN [40]. [40]. Those activation functions have their own characteristics and can be used for different aims and network structures [40]. The non-linear activation functions are classified by their range or the form of the curves [40]. The output of the activation function is defined on the ranging [0,1] or [-1,1].

The sigmoid function is widely used in artificial neural networks [41]. The range of the sigmoid function is [0,1], defined in Equation (2.16) [37]. The function diagram sees in Figure 2.3

$$S(x) = \frac{1}{1 + e^{-x}} \tag{2.16}$$

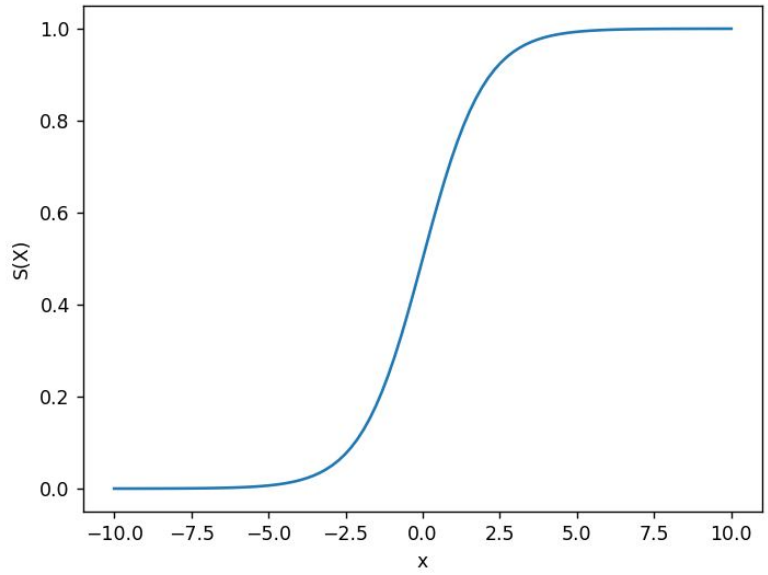


Figure 2.3: Diagram of Sigmoid Function S(x)

The range of a tanh function, also known as Tangent Hyperbolic Function, is [-1,1], defined in Equation (2.19), and the function diagram sees in Figure 2.4 [40].

$$tanh(x) = \frac{sinhx}{coshx} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.17}$$

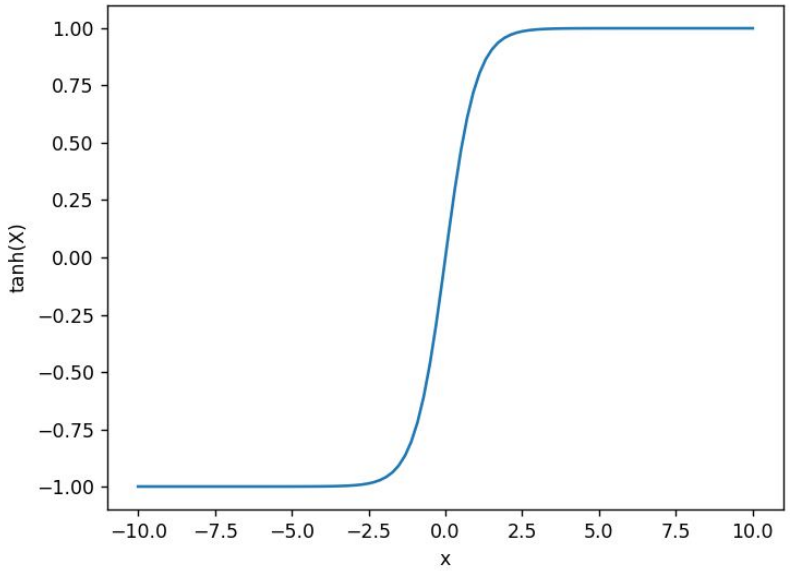


Figure 2.4: Diagram of activation function: tanh(x)

Once a neuron is determined, weights and bias are assigned, signal **y** is passed through the activation function. If the output signal exceeds a given threshold, for the sigmoid function, a neuron is activated that led to the signal passing and **f** is defined by Equa-

tion2.18 [42]. For the tanh function, **f** is defined by Equation (2.19)

$$f(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} \omega_i x_i + b \geq 0, \\ 0, & \text{if f } \sum_{i=1}^{n} \omega_i x_i + b \leq 0, \end{cases} \tag{2.18}$$

$$f(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} \omega_i x_i + b \geq 0, \\ -1, & \text{if f } \sum_{i=1}^{n} \omega_i x_i + b \leq 0, \end{cases} \tag{2.19}$$

Once this neuron is connected to another neuron, the passing data becomes a new input to the connected neuron. Many neurons with associated weights and thresholds connected to each other create an ANN. Once an individual neuron is activated, the data from this neural will be transported to the next neuron. From the inputs of the first level to the input of the last level, a certain number of neurons are combined into different layers. Layers can be classified as the input layer, hidden layer, and output layer [36].

ANN can perform the tasks such as classification, prediction, clustering, and associating [43]. The architecture of an ANN sees in Figure 2.5

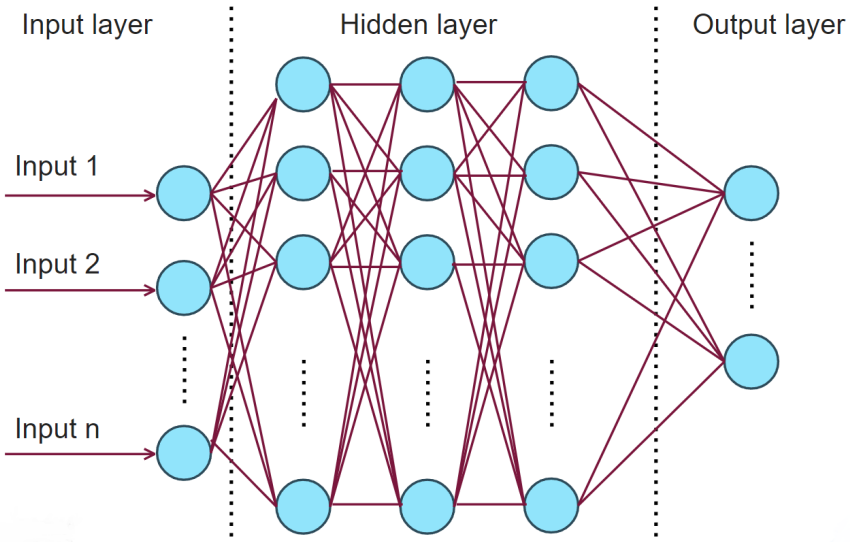


Figure 2.5: The architecture of an ANN [44]

The ANN architecture comprises the input layer, hidden layer, and output layer. The input layer receives the input values and transports the values into the hidden layers, the hidden layer consists of a set of neurons between input and output layers that can be single or multiple layers, and the output layer can include one neuron or multiple neurons [45].

2.4 Speech Synthesis

Speech synthesis is the artificial production of human speech which is a simulation generated by the computer of human speech [46]. Speech synthesis is used to translate a text into spoken speech automatically [47]. The computer system for speech synthesis

is called a speech synthesizer [48]. The speech synthesizer comprises two parts: Natural Language Processing (NLP) unit and the Digital Signal Processing (DSP) unit. The NLP unit handles phonetization, intonation, and rhythm to the input text. The output of NLP is a phonetic transcript that is sent to the DSP unit. DSP unit transforms the phonetic transcript into machine speech [49]. The DSP unit handles the words, phrases, sentences to the actual machine 'pronunciation' that converts the text to human speech articulation [49]. There are two ways to implement the DSP unit: Rule-Based Synthesis and Concatenative Synthesis. The Rule-based synthesizers generate speech via the dynamic modification of several parameters [49]. For concatenative synthesizers, the human speech is generated by producing a sequence of concatenated segments, retrieved from its speech sample database [49].

The process of synthesis process is shown in Figure 2.6.

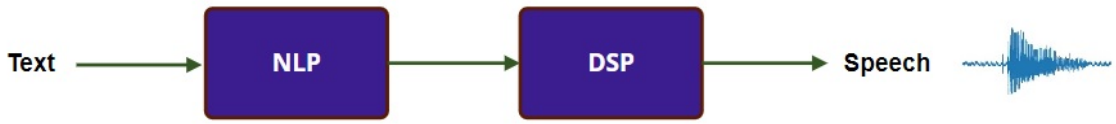


Figure 2.6: Text-To-Speech Synthesis Progress [49]

3

Methods

The prototype of Talk2Me is a communication interface between the ambulance personnel and the communication set that is used in an ambulance for prehospital care. Through the conversation between the user and Talk2Me, the user uses the speech as input for patient **registration** first and the ABCDE assessment. In this chapter, the core methods of building a prototype are described in detail, then the method of test design is introduced.

3.1 ABCDE Assessment

ABCDE Assessment consists of five assessments. Each assessment includes several working steps and each step includes one or more commands combinations that are categorized and the order of commands is fixed. The steps of the ABCDE assessment are shown in Table 3.1 [50].

Setp1	**Step2**
Airway	Clear/Secretions/Occluded
Breathing	Normal/Absent/Cheyne Stoke/Stridor/Shallow
Circulation	Pulse-Strong/Regular/Irregular/Weak
Disability	Speech-Yes/No
	Weakness Face - Right/Left/No
	Weakness Arm - Right/Left/No
	Weakness Leg - Right/Left/No
	Seizure -Yes/No
Exposure	Normal/Abnormal

Table 3.1: ABCDE Assessment working steps

To simplify the command combination, only one keyword is selected that can stand for the meaning of each step and the keyword selection considers also uniqueness. The selected keywords named Command1, Command2, and Command3 (Command3 only for Disability Assessment), create the Two-words commands and Three-words commands respective to the number of the keywords.

For ABCDE Assessment, the Two-words command and Three-words commands are shown in Table A.1 in Appendix A:

.

3.2 Programming Environment

As one of the most popular AI programming languages, Python is used as the programming language in this thesis. The reasons to select Python are as follows:

- Python is easy to understand for its simple syntax, structure, rich processing tools, and large support of libraries and tool-kits. Python's built-in high-level data types and its dynamic typing make the development timeless than other languages [51].
- Python has powerful polymorphic lists and dictionary types [51].
- Python is an object-oriented programming language [52]. Python objects combine data and the method together to make it possible to wrap complex processes [52]. The programmer can pass around python objects instead of data [52]. Through the object reference, the programmer can access any of the attributes of the object.
- Python supplies numerical computing tools, such as Numpy that support n-dimensional arrays. Numpy is open source and easy to use [53].
- Python library **Matplotlib**, supplies powerful data visualization support [54].
- **Kivy** is an open-source Python library for Graphical User Interface (GUI) design to develop multi-platform applications [51]. Kivy supports many devices, such as desktop computers, ios devices, Android devices, and any other touch-enabled professional/home-brew devices supporting tangible user interface objects [55].

Pycharm Community is used as a programming platform. The version used in this thesis is Pycharm 2020.3.5(Community Edition). Pycharm can be used on Windows, macOS, and Linux with a single license key [56]. PyCharm has an interactive Python console and built-in developer tools [56]. It supports multiple scientific packages including Matplotlib and NumPy and Kivy [56].

3.3 Prototype development

The prototype uses the conversation between humans and the machine system that supports the user's completion of NLP to carry out a series of tasks. Speech Recognition and Speech Synthesis supply the technology to make the conversation between people and machines possible. The task of the prototype is to complete user registration with voice commands, perform ABCDE evaluation and information processing. The conversation process is displayed to the user through the GUI for visual communication.

3.3.1 Prototype introduction

The prototype consists of seven nodes: **Start**, **Selection**, **Information**, **Register**, **Mode**, **First**, **Second** and **Third** for user's operation. Function module **ABCDE Handler** handles the voice commands and returns the results to a respective node in turns. The flowchart of the functions that the prototype can achieve is shown in Figure 3.1 and related voice commands are seen in Table A.1in Appendix A. With the aim of improving the practicality, the user can go backward freely through voice commands from current nodes through voice command inputting. The user has permission to correct the old mistake. The system operation can be paused and quit freely. Automated data collection is implemented for the tests.

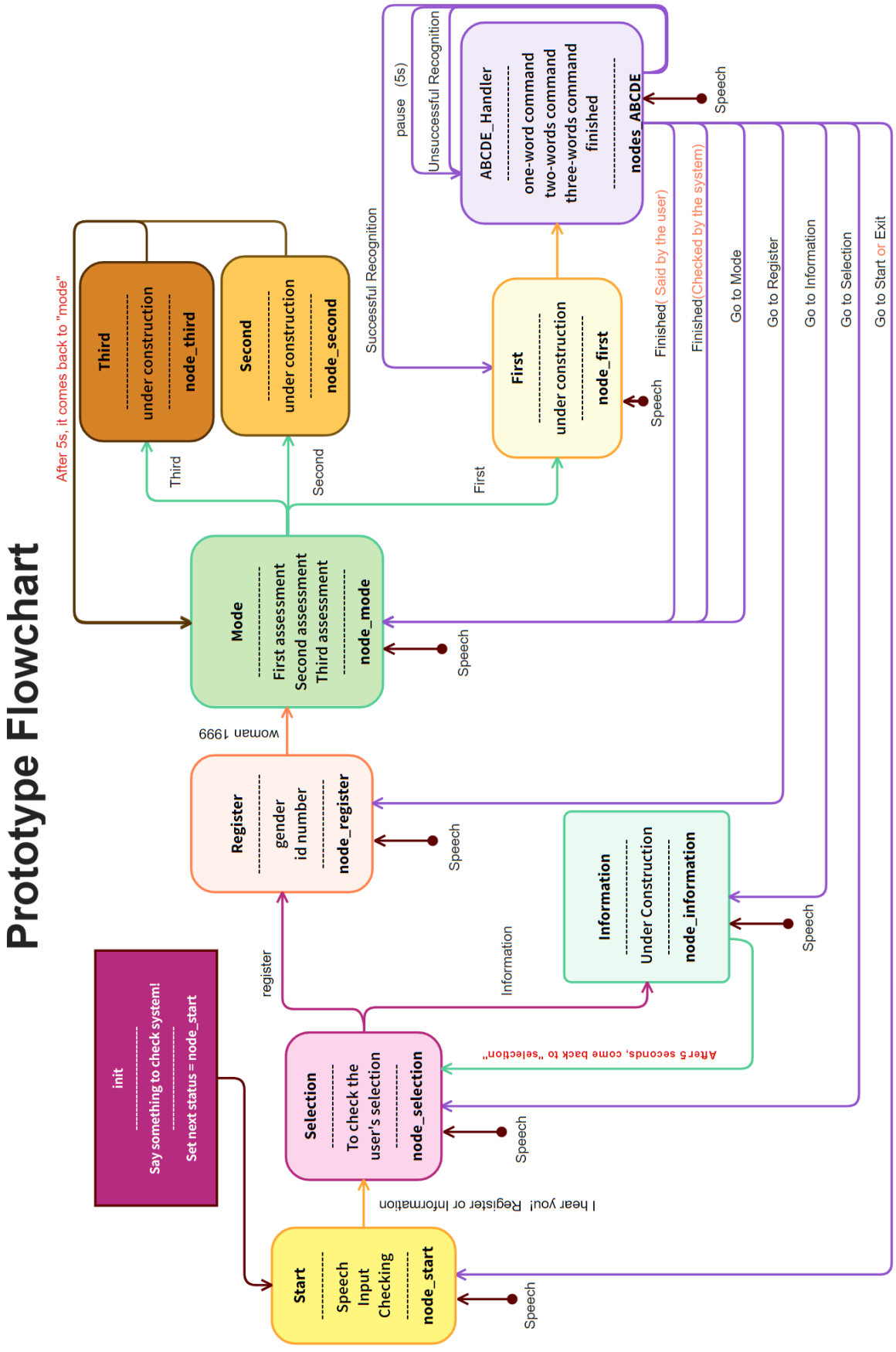


Figure 3.1: The Prototype Flowchart

3.3.2 System Logic Control

A complete conversation consists of a number of dialogues. For each dialogue, it is a sequential communication activity between a human and a machine that creates one programming loop. It is a mixed-initiative interaction between humans and machines [21]. The loops are organized and iterated as the expected conversation. The prototype includes totally seven functional modules **Speech Recording**, **Speech Enhancement**, **Speech Recognition**, **Text Handler** and **Speech Synthesis**. Through **Speech Recording**, the speeches are recorded into audio files as input data to the system. The audio signal with with ambient noise is filtered by **Speech Enhancement**. Next, the filter audio signals are recognized as the texts by **Speech Recognition**. **Text Handler** is designed for text processing to obtain the desired combination of commands and recognition results. Finally, **Speech Synthesis** organizes a dialogue with the user and feedback the recognition result to the user. At the same time, Talk2Me displays the content of the dialogue in real-time on a GUI. The system logic control about one dialogue loop in a conversation is shown in Figure 3.2.

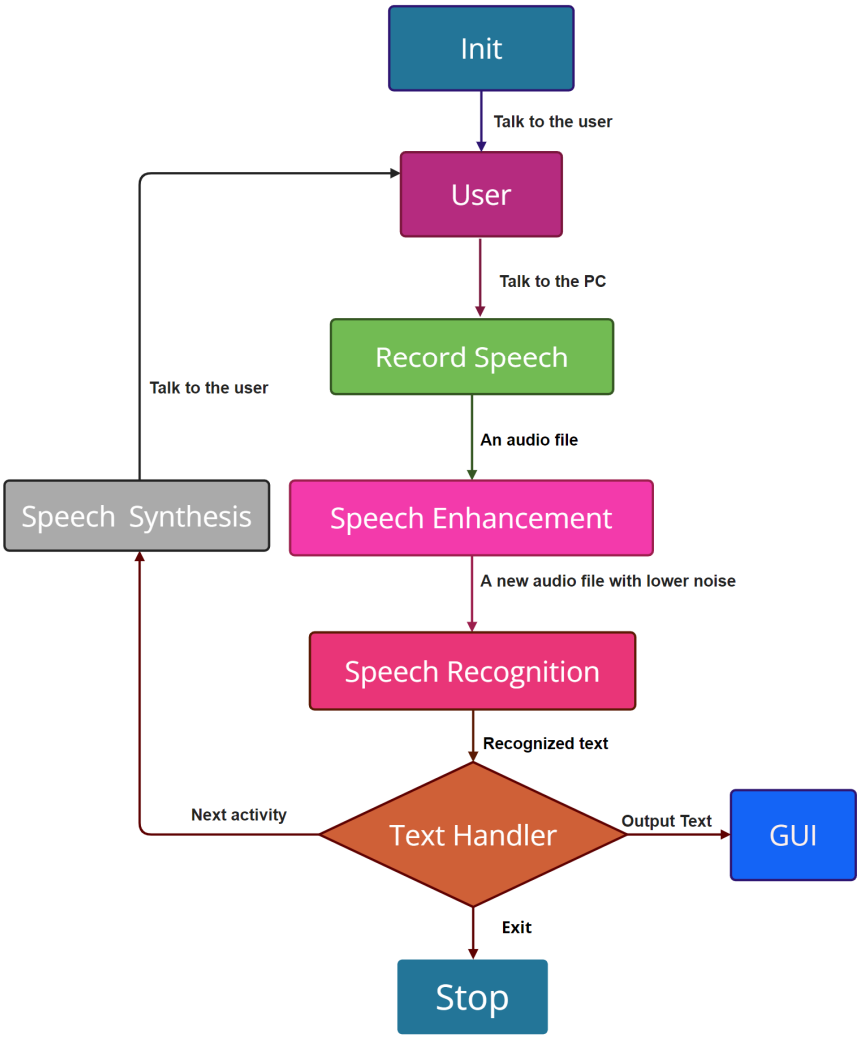


Figure 3.2: System Logic Control

3.3.3 Speech Recording

Objects **Microphone** is created to convert speech to audio through **speech recognition**. The recorded audio files are saved as format .wav.

For Talk2Me, the input is fixed one-word, two-words, and three-word commands. A test about the pronunciation time of the one-word command (yes), two-word command (airways clear), and three-word command(Disability Speech Yes) shown in Table 4.5. The maximal average time for one word is 1.42s, which leads to the time duration of the recording being set to 5 seconds for compatibility with voice commands of up to 3 words.

Commands	Classification	Test1(s)	Test2(s)	Test3(s)	Test4(s)	Test5(s)	Average(s)	Average for one word(s)
Yes	Ordinary speed	1.53	1.01	1.02	1.42	0.99	1.15	1.15
	Slow speed	1.52	1.30	1.05	1.54	1.69	1.42	1.42
Airways Clear	Ordinary speed	1.92	1.79	1.97	1.85	1.27	1.76	0.88
	Slow speed	2.24	2.28	2.14	2.41	2.16	1.81	0.91
Disability Speech Yes	Ordinary speed	2.43	2.74	2.55	2.41	2.35	2.50	0.83
	Slow speed	3.22	3.30	2.90	3.22	3.32	2.61	0.87

Table 3.2: Average pronunciation time for one-word command, two-words command, and three-words command

3.3.4 Speech Enhancement

Speech recognition is greatly influenced by many factors such as the type of noise [57]. A Spectrum-Subtraction-Noise Filter is designed for speech enhancement. The process of noise filtering is first to get the specification of an audio file. For an audio .wav file, the following four important parameters are fixed and can be obtained:

- nchannels: numbers of channels,
- sampwidth: the widths of the sample,
- framerate: the samples rate
- nframes: the number of frames

Speech data as a digital matrix are saved in parameter 'nframes'. The data are sent to the Spectrum-Subtraction-Noise filter for noise handling. The other data, such as nchammels, sampwidth, and nframes, as the single-digit parameters, together with the output of the filter is reformed as a new audio file. This process is the filtering process used in this thesis, see Figure 3.3

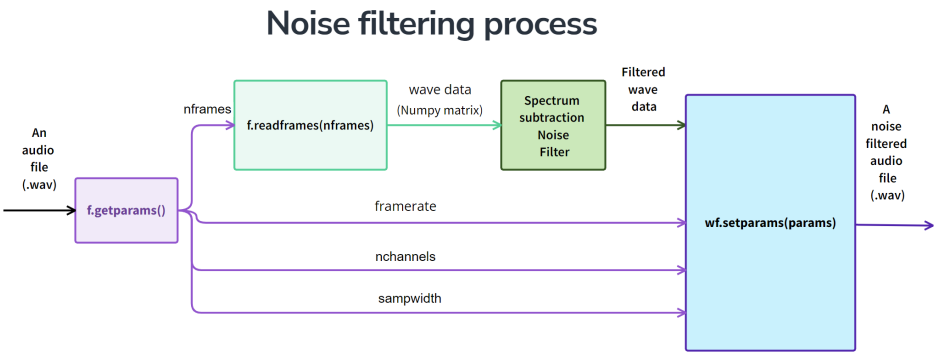


Figure 3.3: The implemented noise filtering process, using a Spectrum-Subtraction-Noise filter

3.3.5 Speech Recognition

There are two ways to achieve Speech Recognition. One way is to use a commercial tool. In this thesis, library **speech recognition** is used. The other way is to create a Neural Network that is more explained in the chapter **Future Work**.

Through **Speech recognition**, object **Recognizer** is created to make **recognize_google** available. The recognition process sees in Figure **myGoogle**

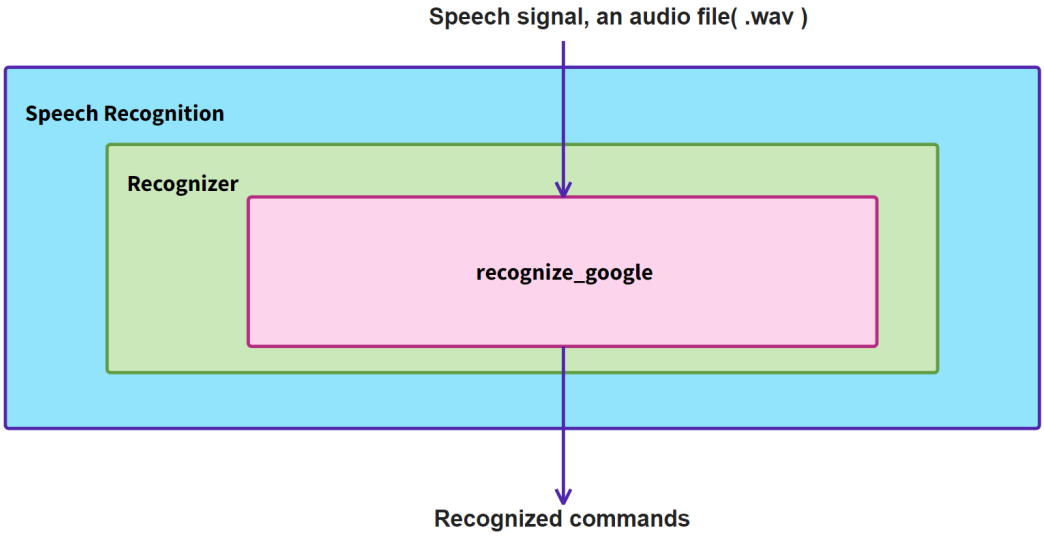


Figure 3.4: The process of speech recognition by **recognize_google** through object **Speech Recognition**

3.3.6 Text Handler

The **Text Handler** is an object. The purpose of the Text Handler is to convert the recognized text into the commands for the iteration control. The input of the Text Handler is the recognized text strings from **Speech Recognition**. The number of the strings in the texts is equal to or larger than zero.

The output of **Text Handler** is a list includes two parts: **OperationFlag** and **resultList**, sees in Table A.4 in Appendix A. **OperationFlag** gives the reference for different nodes, and **resultList** returns the recognized commands in Table A.1.

The structural components of the Text Handler is shown in Figure 3.5. The **Conversion process of input text to a list of strings**, together with three function modules **checkCommand**, **get_Standard()**, **dataCollector** and **checkMatch** inside the Text Handler are introduced.

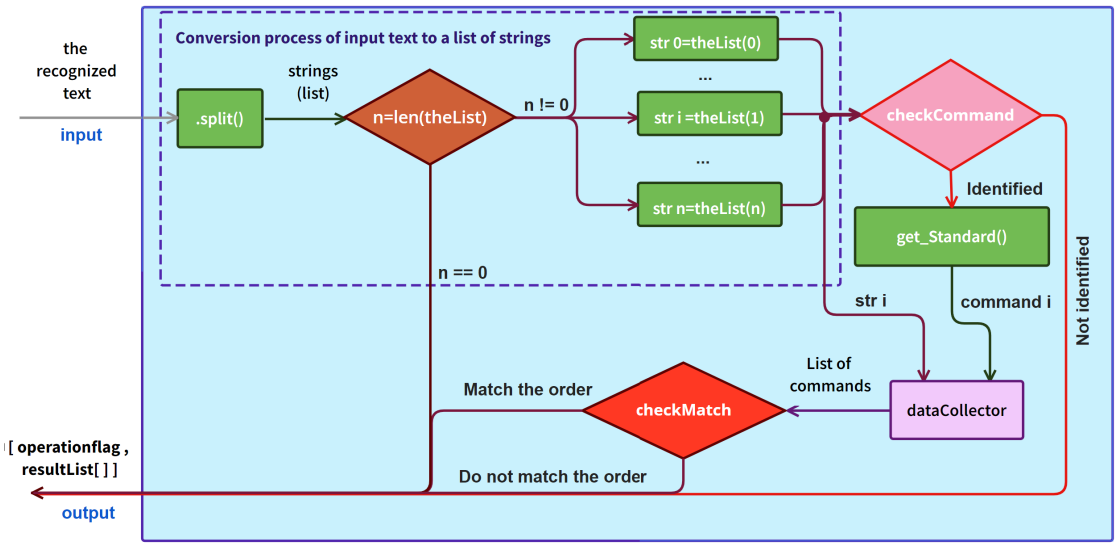


Figure 3.5: Structural components of the Text Handler

1. **Conversion process of input text to a list of strings**

   The input text is the **Recognized text** from the module **Speech Recognition**, see Figure 3.2. The text is first split into strings and stored in a list. The number of the list is equal to or larger than 0. If the number is 0, it means that the recognizer recognizes nothing. The result is sent directly to the output. If the number is equal or larger than 0, all strings in the list are fed sequentially to the next module **checkCommand** through a loop control. This loop control determines that the position of the strings in the list is not important for the next step. It cares not where the string is but if it exists, that makes in a speech, the locations of the commands can be disregarded, and only the correctness of the commands care.

2. **checkCommand**

   Speech Recognize is a public commercial tool that is not designed only for Talk2Me. The range of recognized text from 'Recognize_google' is evanescent. It makes it hard for Talk2Me to recognize the command exactly. As a supplement to this shortcoming, method **Double Expanded Possibility** is designed, seen in Figure 3.6. This method uses two ways to expand the possibility for commands identification. For Way 1, Python class 'SequenceMatcher' compares the input string with similar strings. The output of 'SequenceMatcher' is a digital number that reflects the possibility of the similarity, and the range is [0,1]. If the result is equal or larger than en given ratio, it indicates that the input string is the target, and successful speech recognition is performed. In this thesis, an empirical ratio is selected as 0.7. Way 1 expends the recognition possible for the first time. For the input string, its similar string is not unique. In Way 2, a number of similar strings are saved in a list to be compared in turns with the input string. If any of the comparison results is larger than the given ratio, it is defined as a successful recognition. In this way, the recognition possibility expands the second time.

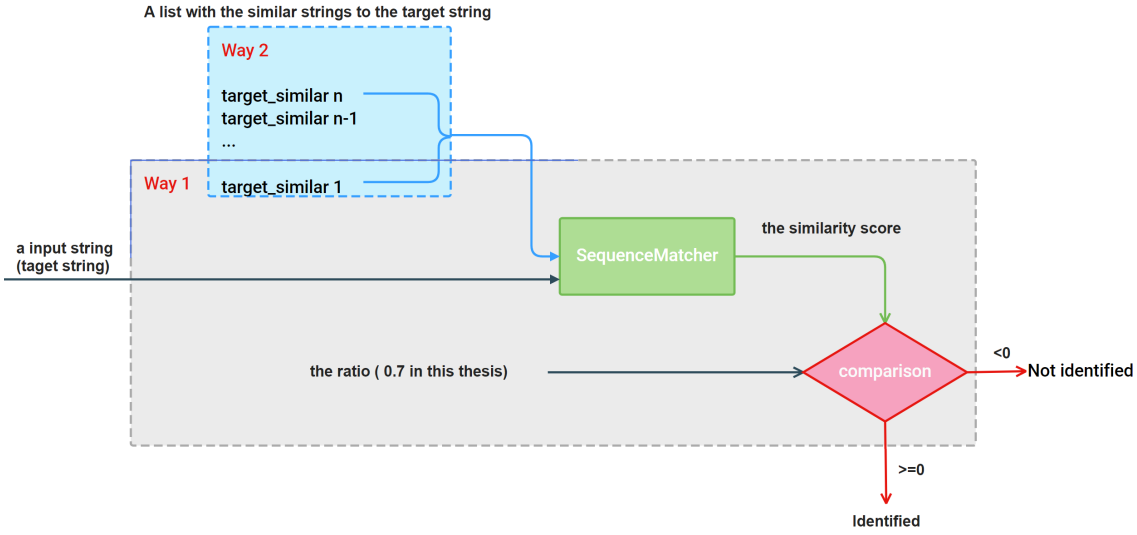


Figure 3.6: The basic module **checkCommand** that uses Method **Double Expanded Possibility** to improve the identification ability

3. **get_Standard()**
   This module is used to get the standard command of the identified string. The standard commands, see Tables A.1 in Appendix A, are stored in their respective lists. These lists, together with the identified string from module CheckCommand are the two input values. The standard command is returned if the input string is in the list. The output is a string. The method to get one of the standard commands of ABCDE Approach is shown in Figure 3.7

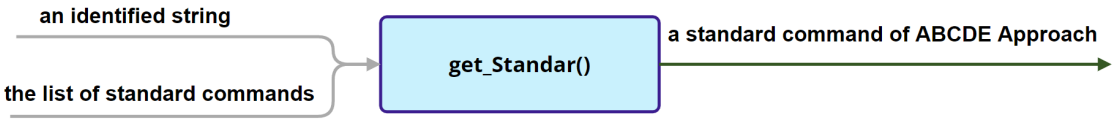


Figure 3.7: A method to get one of the standard commands of ABCDE Assessment

4. **dataCollector**
   It is used to collect the data from the previous loop operation and remove the effect of **Double Expanded Possibility** on certain strings: **normal** and **regular**. There are two inputs for **dataCollector**. One input gets the valid data from $get_Standard()$, and the other input gets the initial strings from the previous loop operation. All data are saved in a new list. For string couples: **normal** and **abnormal** and **regular** and **irregular**, if any of them exists in the list, the second string should be removed. The list is the output of **dataCollector**.
5. **checkMatch**
   The input of module **checkMatch** is a list that includes two or three strings. The strings are a string collection of the results of **get_Standar()**.
   To explain clearly module **checkMatch**, two definations are declared as follows:
   (a) **Commands in the same group**
   In Table A.1 in Appendix A, for commands of ABCDE Assessment, if it exists

a row that includes both command1, command2, and command3(command3 valid only for Assessment Disability), the commands are defined as in the same group.

(b) **Next field of a command**

For the two-word commands, any given command1 corresponds to a unique set of command2, as seen in Table A.1in Appendix A. For the three-word command, it works the same for command1 and command2. They have one-to-one correspondence and do not repeat. The unique set is defined as the next field of the input command.

On this point, the function module **getNextstepField** is designed to get the next field of the given command. The input of **getNextstepField** is a string, the output is the next field of the command. For a list of strings that includes command1 and command2, if command2 is in the same group of command1, then they match. It means that it is successful speech recognition. The logic process is designed in the Figure 3.8. For the three-word command, the procedure is the same by repeating the procedure of Figure 3.8 for two times.

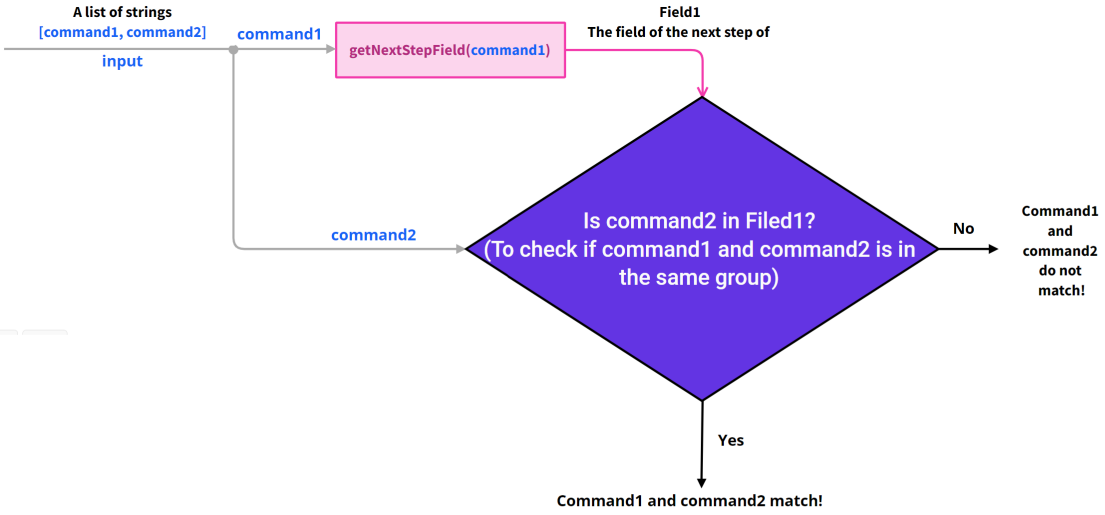


Figure 3.8: The procedure of **checkMatch** to check if the two strings in a list are in the same group

3.3.7 Speech Synthesis

Talk2Me uses Python's library **pyttsx3** [58] for speech synthesis. It is a tool to convert text to speech. An instance object which called **SayIt** is created. The rate of speech talking is controllable through the parameter setting of the instance object. When the rate is set to 150, the speed of the synthesis speech matches the human's delay talking. The higher value of the rate, the faster the speech. In this thesis, to reduce time consumption, the rate is set to 200. Pyttsx3 supplies two kinds of speaking voices. By testings, the second voice has a slight accent, while the first voice is more standard than the second. In this thesis, the first voice is selected for Talk2Me.

3.4 Evaluation Parameters, Variables, and Formulas

This thesis focuses on data related to ABCDE Assessment. In this chapter, three accuracies: ABCDE Input Accuracy, ABCDE Iteration Accuracy, ABCDE Recognition Accuracy together with Speed(the average recording time for one command), are defined as the evaluation parameter used to reflect the system performance.

- ABCDE Input Accuracy: Can't be calculated by the system but guaranteed by the user. For the mistakes that occur during the period of the ABCDE Assessment, Talk2Me gives the user access to correct the mistakes by multiple repetitions of the voice commands. The user can correct the mistakes through multiple repetitions until all the inputs are correct.
- ABCDE Iteration Accuracy: The conversation consists of a number of dialogue loops. A part of loops are of successful speech recognition and fail for the others. This parameter cares about the accuracy based on the times of iteration of loops in a more macro perspective.
- ABCDE Recognition Accuracy: For each loop in the iterations, the number of recognized commands varies from 0 to 3. This parameter cares about the accuracy based on each recognized command in a more micro perspective.
- Speed: The average recording time for one command. The shorter the time, the faster the speed.

The mathematical definitions and other relevant variables and formulas are declared and introduced as follows.

3.4.1 Variables declarations

Variables are declared in Table A.2 in Appendix A. These variables correspond to the corresponding data sources in the automatically stored excel file.

3.4.2 Formula Definitions

The data corresponding to these formulas is not used to directly reflect system performance, but rather as intermediate data for system evaluation parameters.

$$I_{totalTimes} = I_{nothingTimes} + I_{unsuccessfulTimes} + I_{successfulTimesl} \tag{3.1}$$

$$W_{totalWords(min)} \leqslant W_{totalWords} \leqslant W_{totalWords(max)} \tag{3.2}$$

$$W_{missingWords(max)} = I_{nothingTimes} \times 3 \tag{3.3a}$$
$$W_{missingWords(min)} = I_{nothingTimes} \times 1 \tag{3.3b}$$

$$W_{missingWords(min)} \leqslant W_{missingWords} \leqslant W_{missingWords(max)} \tag{3.4}$$

$$W_{totalWords} = W_{recognizedWords} + W_{missingWors} \tag{3.5}$$

$$W_{totalWords(max)} = W_{recognizedWords} + W_{missingWords(max)} \tag{3.6a}$$
$$W_{totalWords(min)} = W_{recognizedWords} + W_{missingWords(min)} \tag{3.6b}$$

3.4.3 Evaluation Parameters

The parameters obtained below are the core of this chapter and this thesis. These parameters are used to reflect the performance of the system, such as recognition accuracy and recognition speed.

$$\textbf{Iteration Accuracy} = \frac{I_{successfulTimes}}{I_{totalTimes}} \times 100\% \tag{3.7}$$

$$\textbf{RecognitionAccuracy} = \frac{W_{recognizedWords}}{W_{totalWords}} \times 100\% \tag{3.8a}$$
$$\textbf{RecognitionAccuracy(min)} = \frac{W_{recognizedWords}}{W_{totalWords(max)}} \times 100\% \tag{3.8b}$$
$$\textbf{RecognitionAccuracy(max)} = \frac{W_{recognizedWords}}{W_{totalWords(min)}} \times 100\% \tag{3.8c}$$

$$\mathbf{t_{AverageTimeFor1Word}} = \frac{t_{DurationTime}}{2 \times W_{totalWords}} \tag{3.9}$$

In a conversation between a tester and Talk2Me, it is always the tester who says the commands first, and then Talk2Me repeats for confirmation. This makes the time used double, which causes factor 2 in equation 3.9.

3.5 Test Design

To test the performance of the prototype, a series of tests were carried out by different test methods. Two types of tests were adopted in this thesis: **Self-Tests** and **Public-tests**. Because of the pandemic, there are 8 testers are selected. The genders of the testers include both males and females. The English level varies from accent to mother tongue. Physical tests and remote tests are taken as the two test methods. For the physical tests, the tester uses the laptop which has installed the prototype. For the remote test, Zoom is selected as the test intermediary. The test backgrounds are multiple from a silent room to the noisy center with high traffic.

A total number of 43 data samples were selected from the tests. 35 samples are provided from the Self-tests. 8 samples are from the 8 testers. The data of the tests come from the first finished ABCDE Assessment. All related result data are saved automatically to an excel file after the tester says "Finished".

3.5.1 Headset used in the testings

For the physical testers, the brand of the headset is **plexgear**, the model **SV-120** [59]. Headset parameters are shown in Table 3.3.

	Item	Value
1	Compatibility	Windows and Mac
2	Drivers	ϕ20 mm
3	Response	20Hz - 20kHz
4	Sensitivity	105 ± 3dB
5	Impedance	20 Ohm
6	Connector	USB
7	Cord length	2.0 m

Table 3.3: Specification of the headset SV-120

3.5.2 Test Methods Design

Two types of tests **Self-tests** and **Public-tests** are adopted in this thesis. The Self-tests were performed by the thesis writer as Tester0 and 8 testers took the Public-tests. The general information about the testers is shown in Table 3.4

Test Type	Tester	Gender	English Level	Accent	Test Method
Self-tests	Tester0	Female	4	Yes	Physical
Public Tests	Tester1	Female	5	No	Remote
	Tester2	Male	5	No	Remote
	Tester3	Male	5	No	Physical
	Tester4	Male	5	No	Physical
	Tester5	Male	4	Yes	Physical
	Tester6	Female	4	Yes	Physical
	Tester7	Female	4	Yes	Physical
	Tester8	Male	5	No	Physical

Table 3.4: General information about the testers

1. **Self-tests** The purpose of Self-tests is to know the performance of the system through a large number of tests under different test environments with different system settings.

   A total of 3 test environments were selected.

   - **E1**: In a quiet room
   - **E2**: In a living room with the TV program playing
   - **E3**: In the center of a small town Gråbo

   A mobile sound meter **examobile**(version 1.2) [60], is installed in an Apple mobile. Noise values were sampled at three locations mentioned above. The sampling time is 10 minutes for each location, see Table 3.5.

Location number	Location	Time(minutes)	Min(dB)	Ave.(dB)	Max(dB)
E1	In a quiet room	10	1	17	67
E2	In a living room with the TV program playing	10	4	38	68
E3	In the centre of a small town Gråbo	10	43	63	72

Table 3.5: Noise values for 3 different environments

For **E3**, Gråbo is a small town close to Gothenburg. There are two supermarkets, several restaurants, shops, a bus station, and several big and small parking spaces in the center, see Figure B.2, The testing time was between 16:00 to 17:30 on a Friday. For the Public-tests, the test locations if of a mixed test environment. The locations were multiple, such as in the living room with the TV program playing, in a garden near the road with a loud motor sound of cars at times

Cases	Test Type	Environment	Ave. Noise Value	With Filter	Tests Times
1	Self-tests	**E1**	17dB	Yes	10
2.1		**E2**	38dB	no	10
2.2		**E2**	38dB	yes	10
3		**E3**	63dB	mix	5
4	Public-tests	A mix test environment		Yes	8
				Total samples	**43**

Table 3.6: Tests arrangement of Case1, Case2.1, Case2.2, and Case3

2. **Public-tests** The purpose of the User tests is for general adaptability. Tests are conducted in a mix of environments with different noise average values. Because the user is expected to be a professional in using Talk2Me, the tester is allowed to familiarize with the system before testing. Tests sample is taken from one of the first two completed tests. The tests instrument is shown in Table. A.3 in Appendix A.

4

Results

The results include parts: The effects of the Spectrum-Subtraction-Noise filter and the data from the designed tests displayed through figures and tables.

4.1 The effects of Spectrum Subtraction Noise Filter

Two types of effects are shown: the input signal is pure ambient noise and the input signal is speech.

4.1.1 Pure ambient noise

The input signal is the ambient noise in location E2 that is shown in the first plot of Figure 4.1. The effect of the filter is shown in the second plot of Figure 4.1.

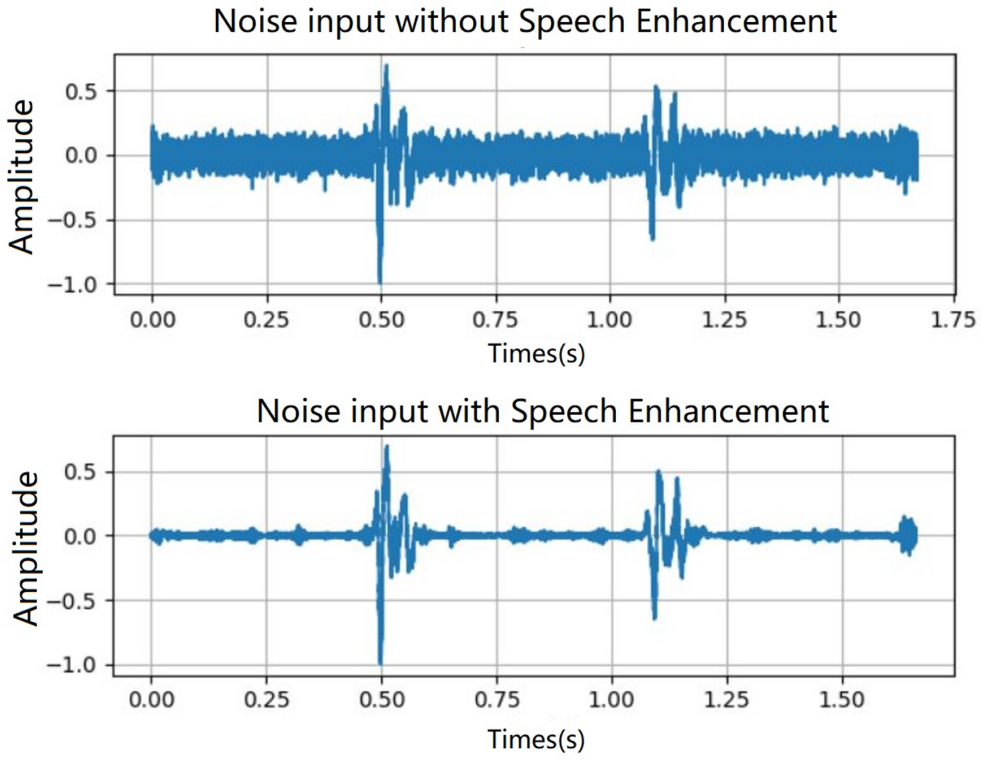


Figure 4.1: Speech enhancement effect: The input signal is the pure ambient noise

4.1.2 A speech

The input signal is a speech in location E2 that shows in the first plot of Figure 4.2. As a comparison, the second plot shows the effect of noise filtering. Noise is filtered and the sound pattern is kept in its original form.

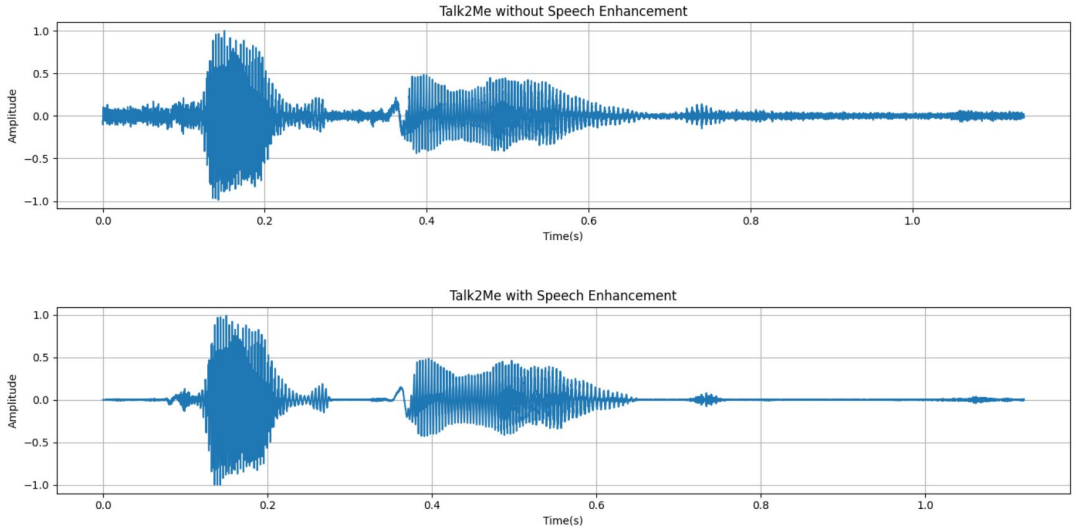


Figure 4.2: Speech Enhancement Effect: Filtering effect on speech with noise

4.2 Results of Self-tests and Public-tests

Three accuracies are cared: ABCDE Input Accuracy, ABCDE Iteration Accuracy, and ABCDE Recognition Accuracy. The ABCDE Input Accuracy is always 100% that is guaranteed by the tester. Each case includes 5-10 independent results shown in Appendix A. The average results for each case are shown as follows:

4.2.1 Self-tests

The results are shown in Tables A.5, Table A.6, Table A.7 and Table A.8 in Appendix A. tests0 includes The results of the evaluation parameters are summarized below in Table 4.1

Item	Test	Location	Filter	Average time for 1 word(s)	Iteration Accuracy	Recognition Accuracy(Min)	Recognition Accuracy(Max)
Self-tests	Case1	E1	No	2.47	89%	84%	94%
	Case2.1	E2	No	2.52	84%	76%	90%
	Case2.2	E2	Yes	2.30	90%	85%	94%
	Case3	E3	No	3.12	86%	77%	91%

Table 4.1: The results of the evaluation parameters for the Self-tests

4.2.2 Public-tests

The results of the Public-tests are shown in Table A.9 in Appendix A. The results of the evaluation parameters are summarized and divided into two groups: Results from the

remote tests, sees in Table4.2, and results from the physical test, sees in Table 4.3.

Item	Test	English Level	Gender	Filter	Average time for 1 word(s)	Iteration Accuracy	Recognition Accuracy(min)	Recognition Accuracy(max)
Public-tests (Remote Tests)	Tester1	5	female	No	11.68	37%	20%	42%
	Tester2	5	male	No	7.56	42%	27%	52%

Table 4.2: The results of the evaluation parameters for the Public-tests, Remote tests

Item	Test	English Level	Gender	Filter	Average time for 1 word(s)	Iteration Accuracy	Recognition Accuracy(min)	Recognition Accuracy(max)
Public-tests (Physical Tests)	Tester3	5	male	No	1.83	97%	95%	98%
	Tester4	5	male	No	2.46	100%	100%	100%
	Tester5	4	male	No	3.93	71%	61%	83%
	Tester6	4	female	No	3.38	79%	70%	88%
	Tester7	4	female	No	2.88	85%	79%	92%
	Tester8	5	male	No	2.47	97%	95%	98%

Table 4.3: The results of the evaluation parameters for the Tester-tests, Physical tests

4.3 Test results of speech recognition for special command arm

During the development and debugging of the software, it was found that few commands in the command table were very hard to be recognized. The success rate of recognition is significantly lower than other commands. A typical example is command **arm**. A test with 30 samples has been done. Totally 30 samples were taken from ten sound sources, see Table 4.4.

Sample Source	Tester	Sampling method	Number of successful recognition	Number of unsuccessful recognition
Real person	Tester0	Physical	1	9
Open source	Commercial voice module	Physical	0	5
Real person	Tester3	Remote	2	1
Audio from real person played by mobile	Anomymizer1	Physical	0	1
Real person	Tester1	Physical	2	3
Real person	Anomymizer2	Physical	0	2
Real person	Tester5	Physical	0	2
Real person	Tester7	Physical	0	2
		Total	5	25
		Accuracy	5/(5+25)*100%=17%	

Table 4.4: For 30 samples of Arm, the recognition accuracy is 17%

4.4 The average pronunciation time for words: yes, airways, clear, disability, and speech

A test about the pronunciation time of a one-word command(yes), two-word command (airways clear), and a three-word command shows in Table 4.5. This result is used as the reference value as the expected true value of reality.

Commands	Classification	Test1(s)	Test2(2)	Test3(s)	Test4(s)	Test5(s)	Average(s)	Average for one word(s)
Yes	**Ordinary speed**	1.33	1.01	1.02	1.42	0.99	1.15	1.15
	Slow speed	1.52	1.30	1.05	1.54	1.69	1.42	1.42
Airways Clear	**Ordinary speed**	1.92	1.79	1.97	1.85	1.27	1.76	0.88
	Slow speed	2.24	2.28	2.14	2.41	2.16	1.81	0.91
Disability Speech Yes	**Ordinary speed**	2.43	2.74	2.55	2.41	2.35	2.50	0.83
	Slow speed	3.22	3.30	2.90	3.22	3.22	2.61	0.87

Table 4.5: Average pronunciation time for one-word command, two-words command, and three-words command

4.5 Comparison between the expected value and test results

To be used in reality, accuracies are required to be 100%. The speed is based on the normal speech speed, see Table 4.5. For the sake of consistency, all test results here are taken from Self-test, from Table 4.1.

Item	Details	Expected Value	Test Results
Accuracy	Input Accuracy	100%	100%
	Iteration Accuracy	100%	71% - 100%
	Recognition Accuracy	100%	61% - 100%
Speed	Average time for 1 word	0.83-1.42 (seconds/word)	2.47 - 3.12 (seconds/word)

5

Discussion

Accuracy and **Speed** are the most concerning points that are quantitatively analyzed through the data of the tests. **Applicability** and **Robust Degree** are discussed more subjectively. **Ethical discussion** that is relevant to the project is discussed in the last part of this chapter.

5.1 Accuracy and Speed

Many factors affect the evaluation parameter **Accuracy**, such as ambient noise, the noise filter.

5.1.1 Self-tests

Case1 and Case2.1, are similar on two points: the testers are the same person and both use no filter. The difference is that the test locations are different, i.e. the value of ambient noise is different. The average noise value for location E1 is 17dB and E2 is 38dB. In Table 4.1, the iteration accuracy of Case1 is 89% and Case2 is 84%, and the recognition accuracy in Case1 is respectively higher than Case2.1. It indicates that **The ambient noise affects the accuracy. The higher value of the noise, the lower accuracy that the system can achieve**. For the recognition speed, the higher the average time for 1 word, the slower speed it has. The value of the average time for one word for Case1 is 2.47s and 2.53s for Case2.2. That indicates that **The ambient noise value affects the recognition speed. The higher value of the noise, the slower speed to recognize a word**.

For Case2.1 and Case2.2, the tester is the same, the value of the ambient noise is the same. The difference is Case2.1 has no noise filter on but Case2.2 has. In Table 4.1, the average time to recognize one word for Case2.1 is 2.53s and Case2.2 is 2.30s. Case2.2 takes a shorter time than Case2.1. And the accuracy values in Case2.2 are higher than those respectively in Case2.1. The comparison between the two cases indicates that **The noise filter can improve recognition accuracy and recognition speed. The speech enhancement works.**

For Case1, Case2.1, and Case3, all of them have no filter. Case1 has the lowest noise value 17dB and the shortest average time 2.47s. Case3 has the biggest noise value of 63dB and has the longest time of 3.12s. The difference between Case1 and Case3 is $3.12 - 2.47 = 0.65s$. That indicates that **The ambient noise impacts the recognition speed significantly.**

5.1.2 Public-tests

The shortest time for public-tests is 1.83s from tester3, Table A.9 which is quite closed to the biggest reference value 1.42s, Table 4.5. In terms of this data, this thesis provides a relatively successful prototype. Tester3, tester4, and teser8 speak English standard English. Their English level is assessed at 5. In Table 4.3, their accuracies are all above 90% and the average times are shorter than 5s. For the other users, their English level is 4. Their average times are all longer than 5s and the accuracies are between 70% to 90% that lower than tester3, tester4, and tester8. From this perspective, it indicates that **Those who speak English well, have better accuracy and shorter average time.** For the gender, three of the testers are female Their average time is longer than the males', and accuracy are lower than 90%. It seems that the female has a worse result. But their English level is all on level 4. There are no three female testers with the same English level of 5, which causes no data to be compared with tester3, tester4, and tester5. It makes that **It is hard to make a conclusion if gender affects recognition performance.** Data in Table 4.2 shows that for the remote tests, the accuracy is low and speed is slow. It indicates that remote test is not the right way for the Public-test. During the pandemic period, the inability to test remotely reduced the number of testers significantly.

5.2 Other factors that affect Accuracy

From Section 3.4.2, in Equation (3.7) and in Equation (3.1):

$$\textbf{Iteration Accuracy} = \frac{I_{successfulTimes}}{I_{totalTimes}} \times 100\%$$

$$I_{totalTimes} = I_{nothingTimes} + I_{unsuccessfulTimes} + I_{successfulTimesl}$$

For a test, $I_{totalTimes}$ is a fixed number. Then **Iteration Accuracy** is only determined by $I_{successfulTimesl}$, that can be rewritten as follows:

$$I_{successfulTimesl} = I_{totalTimes} - I_{nothingTimes} - I_{unsuccessfulTimes}$$

Then **Iteration Accuracy** is updated as follows:

$$\textbf{Iteration Accuracy} = \frac{I_{totalTimes} - I_{nothingTimes} - I_{unsuccessfulTimes}}{I_{totalTimes}} \times 100\%$$

The solution to increase the value of **Iteration Accuracy** is to decrease the value of $I_{nothingTimes}$ and $I_{unsuccessfulTimes}$. The factors that can affect $I_{nothingTimes}$ is as follows:

1. **The factor that affects** $I_{nothingTimes}$ In Table A.2, $I_{nothingTimes}$ is defined as "Numbers of the iteration which "Record Nothing" ".Here, "Record Nothing" does not mean that nothing is recorded but the recognizer can't recognize the input audio signal to a text. There are many reasons for this problem to occur, and the main reasons are analyzed as follows:
   (a) **The speech recording is out of sync with the user's voice commands** The recorder records the sound for the time period before or after the start of the speech. It means that the recorded input signal is only the ambient noise.

The user speaks too early or too late that causes only the noise recorded. The rhythm of the human side of the conversation in human-machine communication affects the test performance of the system. **That speech can be recorded in time and supply valid input data, is the base for speech recognition.**
Solution:

- A reminder sign is provided on the user interface to inform the status of the system: recording or the end of the recording.
- The user uses more time to be family with the rules of Talk2Me. Control the pace of speech. To speech and reply at the proper time.
- Algorithm optimization for recording, see Future Work.

(b) **Noise affects the accuracy** Ambient noise seriously affects the recognition accuracy.
Solution Use Kalman filter to improve the speech enhancement, see Future Work.

(c) **Special words require extremely accurate pronunciation such as arm**. In Table 4.4, the accuracy of the successful recognition in 30 samples is 17%.
Solution:

- The testers correct their pronunciation.
- The reason that causes the problem is still uncertain. Need to do more research work in the future if necessary.
- To use an ANN for speech recognition. It might not be a problem for an ANN. Optimize the program and add user recognition to the system to enhance recognition. Collect samples from the testers, and those special words are no longer "special" any longer. See **Future Works**.

(d) **Bad headset The quality of the headset affects the accuracy greatly.** Two headsets were used during the debugging period. Before the midterms evaluation, an old simple headset to a Huawei mobile was used. The possibility of **"Record Nothing"** happened quite often. It is not strange that in many cases, it happened more than 50% in a test. After the midterms evaluation, a new headset replaces the old one. The probability of Recording nothing significant declines.
Solution Select the right headset and use ANN for speech recognition. See Future work.

(e) **Internet constraints system operation and decline the accuracy**
Talk2Me uses the online tool recognize_google for speech recognition. The condition of the internet greatly affects accuracy. From Case4, the possibilities of Recording Nothing for tester1 and tester2 are respectively $64/121 = 53\%$ and $18/115 = 10.3\%$. Both are higher than the other testers who take the physical tests, and the reason is uncertain.
Solution:

- If the recognition method is the only choice for speech recognition, it is necessary to find the reason. To find a better software platform and optimize the algorithm.
- Use off-line tools for speech recognition.
- Use an ANN for speech recognition

2. **The factor that affects** $I_{unsuccessfulTimes}$ The reasons that cause unsuccessful recognition are multiple.
   (a) The algorithm is not good enough to recognize right.

   Double Expanded Possibility is used to increase the recognition ability. For most of the commands, it works well. But this algorithm is not perfect that can cause recognition mistakes.

   - In this thesis, there are two pares of words: **Normal, Abnormal**, **Regular Irregular**. The recognition mistakes occur because the identification areas overlap. Figure 5.1 illustrates how an overlap creates between Normal and Abnormal.

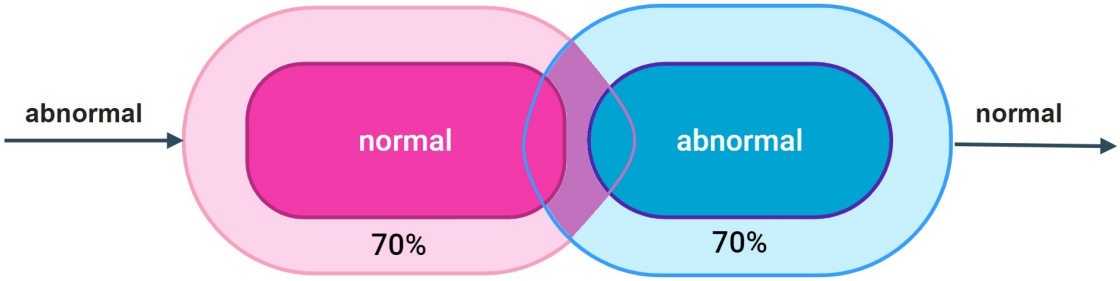


Figure 5.1: An overlap occurs between **normal** and **abnormal**. A recognition mistake occurs when the input text is: "abnormal" and the recognized command is "normal"

Solution Find out the special commands with the overlap problem during the debugging period. Save them in a list and remove the effect of method **Double Expanded Possibility** for the commands inside the list. The solution shows in Figure **solutionFornormal**

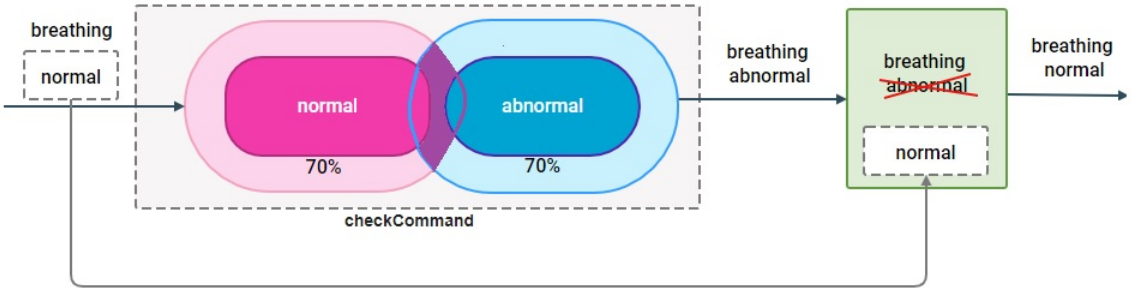


Figure 5.2: The method to remove the method **Double Expanded Possibility from special commands**

- Similar words cause mistakes. For tester 1, the input command is "seizure", google_recognition recognizes it as "Fisher". "Fisher" is in the expanded range of command "finished". It causes recognition mistakes, see Figure 5.3.

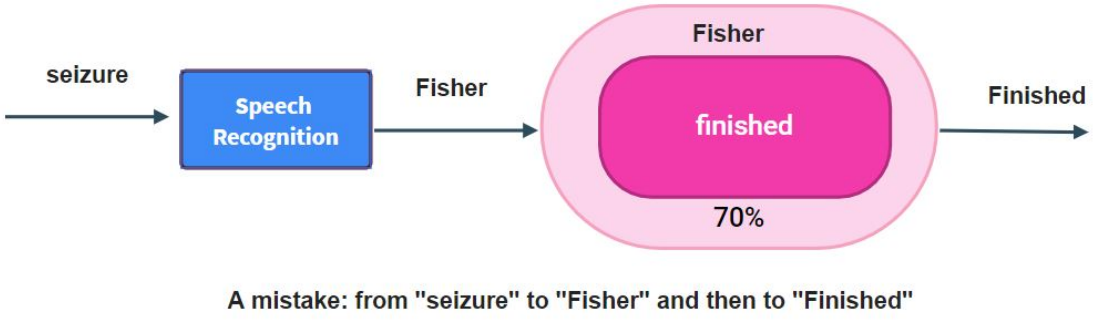


Figure 5.3: Double Expanded Possibility causes mistakes from "seizure" to "Fisher" and then to "Finished"

Solution: Algorithm optimization.

(b) For certain people, their English has an accent and the pronunciation is vague and difficult to identify. Special words are difficult to be recognized, such as "arm".
Solution:
- Algorithm optimization
- Use ANN for speech recognition

(c) Similar words are hard to be identified and recognized right. There are too many words in the English vocabulary that have the same or similar pronunciation.
Solution Use ANN for speech recognition

(d) The user's input mistake. Inputs from the tester do not match the expected input. The combination of input commands is fixed. The user's voice input is outside the range of the combination. The voice command is successfully identified, but outside the range of the combination.
Solution
- The user uses more time to learn the command regularly.
- A remind sign on the user interface to show which commands combination can be selected.

Solution
- Optimization of voice recording control.
- Optimization of the speech repetition process.

5.3 Applicability and Robust

In English, the definition of **Applicability** is "the fact of affecting or relating to a person or thing" [61]. For Talk2Me, it means whether the design of the software is suitable for people's usage habits and whether it is universal for different users. Talk2Me has been designed with a logical and rational approach, starting from the actual requirements. The software provides patient registration, modality selection, information management, and ABCDE assessment input management. A human-machine dialogue and real-time feedback are used to accomplish the tasks. The software also gives the user a lot of

flexibility to switch between different functional modules. The process design sees in Figure 3.1. The design process of Talk2Me is suitable for people's usage habits.

In test Case4, 8 testers have taken the tests. In the results of the sex physical tests, three of them have iteration accuracy: 97%, 100%, and 97%, and the other three are at the same level as the General Tests. It shows that the results of the Tester Tests are better than the results of test0 and the program is not designed only for the thesis writer: test0. It indicates that Talk2Me is universal for different users.

For the **Robust** degree, **Both internal and external factors affect the stability of the system.** For the internal factor, there are still bugs that existed in the prototype. Minor bugs can cause the software to enter a dead loop, and serious bugs can cause the software to crash and become unusable. The more debugs that exist in the prototype, the more unstable the program is and the worse Robust degree the prototype has. Because of time constraints, not all the logic bugs are found and solved in this thesis. In special conditions, the program does not run properly and it occurred in a few of the test cases.

External factors affect the robust degree too. Talk2Me uses the online tool for speech recognition that determines Talk2Me requires both Internet service and remote service from the tool's provider. If no internet, Talk2Me can't work. If the tool's provider stops the remote service, Talk2Me can't work. If the tool provider provides defective technical support, it can affect TALK2me.

There are two remote tests by tester1 and Tester2, see Table 4.2. For tester2, the Iteration Accuracy is 42% and the average time for one word is 7.56s. Tester2 has taken a similar test in the midterm evaluation. The test result is good. The recognition accuracy is in the range of 90%-95%, see Table A.10 in Appendix A. From 42% to 90%-95%, it is a big difference. The average time is 7.56s which is more than 6 times, ($(7.56 - 1.42)/1.42 = 6.14$), slower than the ordinary time of 1.42s). Here 1.42s is the shortest average pronunciation time of word **yes**, see Table 4.5.

This reason makes that **If an online tool is a part of the program, the program is not only dependent on local software and hardware but also subject to external conditions. The robust degree declines**.

5.4 Ethical Discussion

In recent years, artificial intelligence, also as AI, has been widely used in many fields of life [62]. While we enjoy the convenience, AI has also attracted great attention to its impact on society [62]. Speech recognition as part of AI, the source of data, privacy are considered in this project. AI technology can cause privacy problems. It impacts the privacy of individuals [63].

In this thesis, once the prototype was finished, volunteers were sought to test the performance of the system. How to find them, what kind of information needs to be prepared, and what problems occurring in the test need to be realized for consideration and preparation. In addition, the test also needs to consider the public interest and design the test in such a way that it does not affect them. For the General Tests, Case 3 was done in a public place. The impact of the tests carried out on the public environment is also being considered. As an NLP and AI project, speech recognition needs lots of speech samples. In this thesis, a test on speech recognition was done. Thirty speech samples of

arm need to be collected. The way and manner in which the samples were obtained should be considered. For the analysis of the results of these tests, private information such as the name of the testers cannot appear in this thesis, the testers should be protected and anonymized otherwise it is a privacy violation to the sample provider. After the project is finished, how to handle those data(audio files) is another problem. To save them as an NLP resource for future usage or to erase them immediately should be two alternatives.

6
Conclusion

The purpose of Master Thesis Talk2Me is to establish a user interface to complete patient registration and ABCDE Assessment and then to test the software performance to evaluate whether real-time documentation and information handling is a suitable method used in ambulances during acute situations. The prototype supplies the user with an interface to complete patients' registration and ABCDE Assessment. There are many factors that affect the performance of the prototype, such as the ambient noise, the method of implementation of speech recognition, speech enhancement, headset used, and even the user's oral English level. Among these factors, the most important parameter affecting the performance of the system is the speech recognition method adopted by the system. A good way is to use ANN, however, due to the lack of voice training data, ANN can not be selected. This thesis uses a ready-made commercial tool Python Library 'SpeechRecognition' for speech recognition.

For a system that uses a commercial tool for speech recognition, Talk2Me reaches the iteration accuracy between 70%-100%. Through repetition of the voice commands input, the user can correct systems mistakes to reach the Input Accuracy of 100%. The average time to recognize a voice command is between 2.47s-3.12s per command and the minimum value is 1.83s which is rather closed to the maximal expected value 1.42s. Talk2Me is ultimately intended for use in ambulances, where 100% accuracy and fast recognition speed are required. From this perspective, for the real-time documentation and information handling, there is still a gap between the results obtained in this thesis and the actual demands.

The prototype from this thesis now does not suit to be used directly in the ambulance. but as a transitional pre-test software, it is successful. The prototype designed from this thesis is preferable. A method that can improve accuracy and speed is needed to achieve the real-time control requirements. There are two alternatives as to the solution. The first option is to continue to use the commercial tool, to find a better alternative tool, or improve the usage of the selected commercial tool to achieve the expected results. The other alternative is to abandon the commercial tools and use ANN for speech recognition. A number of suggestions for specific solutions in the chapter Future work are introduced. If the recommended future work is done in the future, Talk2Me can be actually used in reality. Real-time documentation and information handling during acute situations is a suitable method used in ambulances.

7

Future Work

In the future, a number of tasks will need to be done to complete the prototype. At the beginning of this chapter, it introduces the measurements that can significantly improve the performance of prototype, then follows the possible work around **Accuracy**, **Speed**, **Robust**, **Applicability** and **Tests**.

7.1 Important Work Initiatives

The following recommended works can significantly improve the applicability of the system.

1. **Using ANN can greatly improve speech recognition** Abandon the existing speech recognition method provided by the commercial tool. A self-built ANN should be used instead. To obtain an optimized system, for a self-build ANN, enormous training data are needed to train the system.
   To create an AI neural network, the logical implementation seen in Figure 7.1

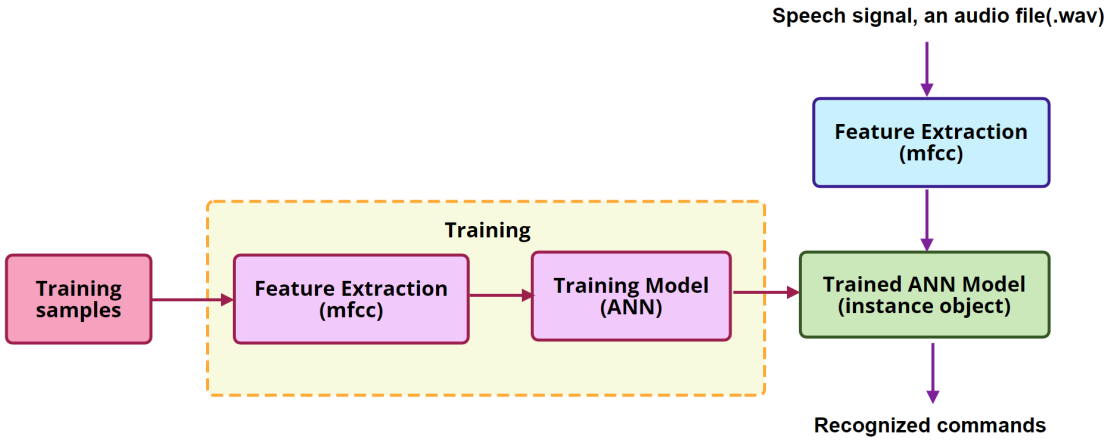


Figure 7.1: Speech recognition from ANN

To obtain the optimized system, for a self-build neural network, enormous training data are needed to train the system. Training an Artificial Neural Network is an iterative Learning Process. For Talk2Me, an NPL project, the training samples are a set of audio files of commands. When audio is fed into the system, the first is to obtain the audio feature through Mel Frequency Cepstral Coefficents (MFCC) [64]. MFCC is the technique to extract the features from the audio signal. The data then is sent to the ANN to be learned with the associated weight. The network's

calculated values for the output can be signed as 1 if it is a correct value, otherwise, 0 if is not correct. The difference between the calculated value and the correct value is the error that can be used for the next iteration for weight adjustment. After all the samples are learned then a trained ANN model is created as an instance object that is used for speech recognition.

2. **Speech recording should be controllable** In this thesis, speech recording is uncontrollable. The longest recording time is 16s in Case2.1. Need work future to improve and optimize the process to make recording controllable.
3. **Build a hybrid hand-touch and voice-activated control app**
   - Hand-touch control: Available for system maintenance and other situations that do not require to be used in the ambulance.
   - Voice-activated control: Only use in the ambulance.
4. **Special identification to the user** A user-specific sampling function is created to capture the user's voice characteristics, allowing talk2me to become familiar with the user's voice characteristics.
5. **Convert Talk2Me from a PC App to a mobile App/Android APK** The prototype is used in the ambulance. It should be installed in mobiles, headset Realwear HMT-1, or other hardware. More works should be done in the future to convert the prototype from the windows system to multiple operating systems, such as Apple iOS, Android.

7.2 Accuracy

The Kalman filter should be completed. The effect of speech enhancement needs to be expressed in data for quantitative measurement of the filter's effect on system accuracy. In order to improve the noise filtering effect, a Kalman filter is expected to be serially connected to the Spectrum Subtraction Filter, see Figure 7.2.

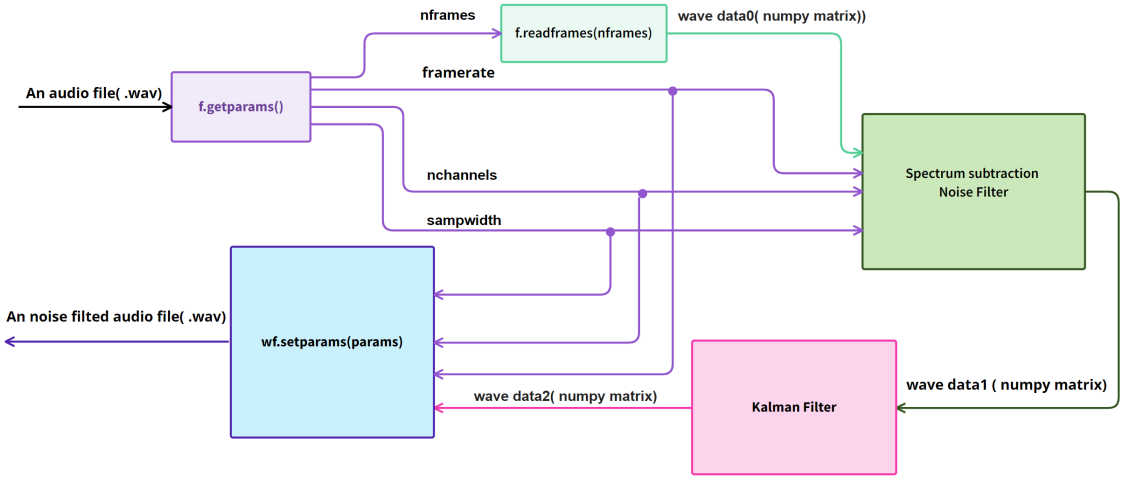


Figure 7.2: A Spectrum Subtraction Filter and a Kalman filter are series-connected for better speech enhancement.

From Figure 7.2, wave data1 is the output of the Spectrum-Subtraction-Noise filter

that is the observation values to the Kalman filter. wave data2 is the estimated value of the Kalman filter.

A Kalman filter is expected to be serially connected to it but not actually used in this thesis for the low operating speed. But this thesis shows how well it works. The effect of Kalman filtering on 100 samples taken from a random audio file is shown in Figure 7.3.

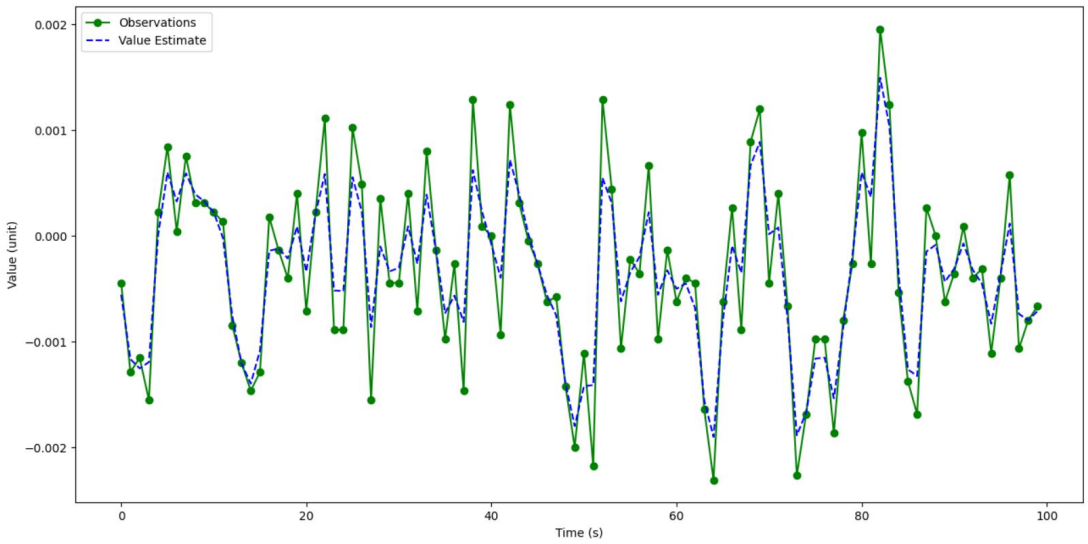


Figure 7.3: Kalman filter, To be done in the future works.

7.3 Speed

- **The method for speech repetition needs to be improved** In this thesis, after the recognition of the input speech, repetition to the recognized commands tells the user about the recognized results. It makes the speed double. A better optimization approach needs to be adopted to reduce the repetition time.

7.4 Robust

- **Structure Optimization:** Because the time limitations, the prototype is not 100% perfect. There are still bugs. The structure of the prototype should be optimized.

7.5 Applicability

- **Obtain first-hand information from ambulance medical nurses** More communication work with the ambulance nurses to get the actual requirements.
- **Add more assessments** This master thesis focuses only on the first assessment(ABCDE Assessment). More useful assessments should be implemented.
- **Information** Function block "Information" should be finished! Can do jobs like searching, deleting, summarizing the patients' data, and other supporting work, such as printing, PDF file generation, etc.

- **GUI Construction :** More widgets should be added to make the prototype with better practices.
  - A **TabbedPanel**, see Figure 7.4, can be used to classify First Assessment, Second Assessment, and Third Assessment.

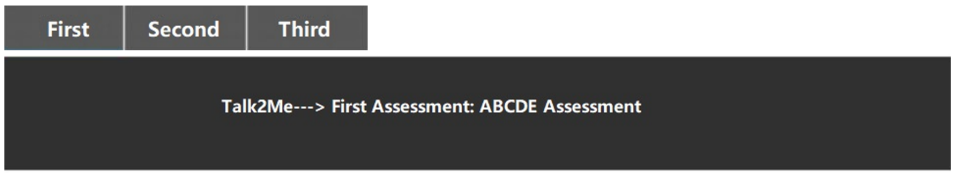


Figure 7.4: A **TabbedPanel** can be used to classify First Assessment, Second Assessment, and Third Assessment

  - A **Slider**, see Figure 7.5, can be used to graphically show the degree of completion of each assessment.

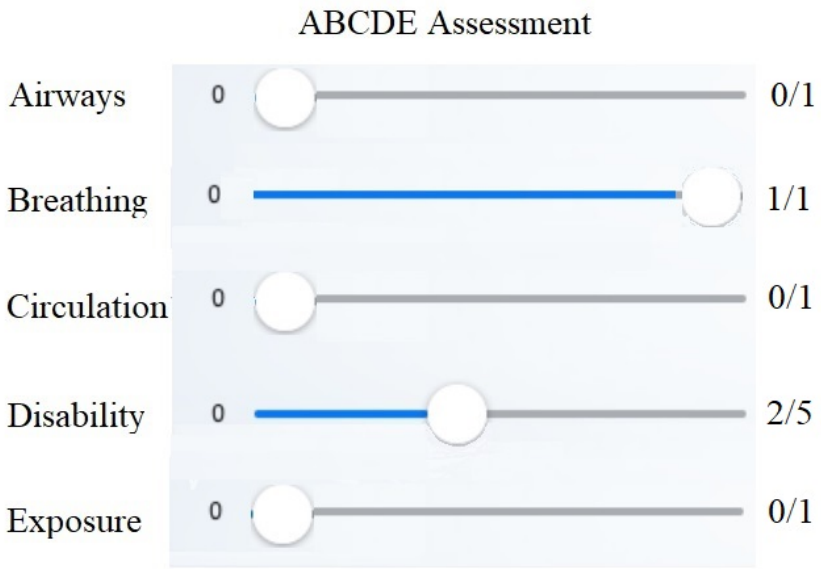


Figure 7.5: A slider can be used to graphically show the extent to which the ABCDE Assessment has been performed

- **Data communication** Implement the Wireless Data communication between **Talk2Me** and the hospital's data center

7.6 Tests

More tests should be done on different occasions, such as in the ambulance, a running car, etc.

A

List of Tables

A.1 Tables for chapter Methods

Classification	Command0	Command1	Command2	Explanation
System Commands	exit	/	/	Exit the program
	finished	/	/	To end the First Assessment and give command to save the data for test.
	first	/	/	Select the node First
	information	/	/	Select the node Information
	go	to	register	Three-word commands Go to the node that selected.
	go	to	selection	
	go	to	start	
	pause	/	/	Pause the systme for 5 seconds
	register	/	/	Select the node Register
	second	/	/	Select the node Second
	third	/	/	Select the node Third
ABCDE Approach	airways	clear	/	Command0 can be an one-word command Command1 can be an one-word command Command0 + Command1 is a Two-words command
		secretions	/	Two-words Command
		occluded	/	Two-words Command
	breathing	normal	/	Two-words Command
		absent	/	Two-words Command
		stokes	/	Two-words Command
		stridor	/	Two-words Command
		shallow	/	Two-words Command
	circulation	strong	/	Two-words Command
		regular	/	Two-words Command
		irregular	/	Two-words Command
		weak	/	Two-words Command
	disability	speech	yes	Command0 can be an one-word command Command1 can be an one-word command Command2 can be an one-word command Command0 + Command1 + Command2 is a Three-words command
			no	
		facial	right	
			left	
			no	
		arm	right	
			left	
			no	
		seg	right	
			left	
			no	
		seizure	yes	
			no	
	exposure	normal	/	Bit0 or Bit1 can be an one-word command Bit0 + Bit1 is a two-word command
		abnormal	/	

Table A.1: Command table of Talk2Me

SN.	Item	Description
1	$I_{totalTimes}$	Total numbers of the iterations.
2	$I_{nothingTimes}$	Numbers of the iteration which "Record Nothing".
3	$I_{unsuccessfulTimes}$	Number of the user's speech is recognized partly or totally to wrong commands.
4	$I_{successfulTimes}$	Number of the user's speech is recognized right.
5	$W_{totalWords}$	Total number of the words the user says. Talk2Me can't get exactly this data because for each "Record Nothing", Talk2Me records only the noise. Talk2Me does not know the numbers of the user says. But the range of values for this data is known: 1, or 2 or 3.
6	$W_{totalWords(max)}$	In a single conversation, for all those "Record Nothing", the number of words of each iteration is 3.
7	$W_{totalWords(min)}$	In a single conversation, for all those "Record Nothing", a number of words of each iteration are 1.
8	$W_{recognizedWords}$	Number of the words recognized successfully inside the set of confirmed commands.
9	$W_{missingWords}$	The actual number of words that the user says which are totally missed by Talk2Me. It is uncertain.
10	$W_{missingWords(max)}$	Total number of the words when "Record Nothing" happens. For those "Record Nothing", the numbers of the user's input commands are 3.
11	$W_{missingWords(min)}$	Total number of the words when "Record Nothing" happens. For those "Record Nothing", the numbers of the user's input commands are 1.
12	IterationAccuracy	Accuracy to the effect of iteration of the conversation.
13	RecognizeAccuracy	Accuracy to reflect the user command input identification
14	$RecognizeAccuracy_{(max)}$	Maximum likelihood of RecognitionAccuracy. For all "Record Nothing", the number of missing words that the users input is 1.
15	$RecognizeAccuracy_{(min)}$	Minimum likelihood of RecognitionAccuracy. For all "Record Nothing", the number of missing words that the users input is 3.
16	$t_{DurationTime}$	Total time for the ABCDE Assessment
17	$t_{AverageTimeFor1Word}$	The average operation time for one word consumed in the conversation between human and Talk2Me.
18	$t_{RecordingTime(max)}$	Maximum time of recording one word in one conversation
19	$t_{RcordingTime(min)}$	Miminum time of recording one word in one conversation
20	$t_{RcordingTime(ave.)}$	Average time of recording one word in one conversation

Table A.2: Variable declarations

Test Instrument of Talk2Me

Item	Explaination		Commands			
			Command1	Command2	Command3	
Step 1	Check System, you can say anything.	C1	Hi hi, Talk2Me!			
Step 2	Select **Register** or **Information**. You should select Register.	C2	Register	/	/	
Step 3	Say **Sex (**woman, man, male, female) and the **year** (4 numbers)	C3	Woman	1999	/	
Step 4	Confirmation, you can say **Yes** or **No**	C4	Yes	/	/	
Step 5	Say **First** to selection First Assessment (ABCDE Approach)	C5	First	/	/	
Step 6	Rules for commands (C6-C27) are as follows:	C6	Airway	Clear	/	
	1) The rule for each row of commands is same.	C7	Airway	Secretions	/	
	2) For each new command, you can start from a whole row:	C8	Airway	Occluded	/	
	(Command1 + Command2)	C9	Breathing	Normal	/	
	or	C10	Breathing	Absent	/	
	(Command1 + Command2 + Command3)	C11	Breathing	Stokes	/	
	3) If 2) recognition failed, you can repeat it again	C12	Breathing	Stridor	/	
	4) If 2) is partly recognized, you can only say one command	C13	Breathing	Shallow	/	
	at a time for the remaining ones that weren't recognized	C14	Circulation	Strong	/	
	One example:	C15	Circulation	Regular	/	
	For command C18 **(Disability Speech Yes)**	C16	Circulation	Irregular	/	
	Iteration 1:	C17	Circulation	Weak	/	
	You said: (Disability Speech Yes)	C18	Disability	Speech	Yes	
	She replies: (Disability Speech. Next!)	C19	Disability	Speech	No	
	Iteration 2:	C20	Disability	Facial	Right	
	You said: (Yes)	C21	Disability	Facial	Left	
	She replies: (Disability Speech Yes. Next!)	C22	Disability	Facial	No	
	5) For each finished input, if you want to correct it,	C23	Disability	Leg	Right	
	you can make a new input as 2) (See Line C8)	C24	Disability	Leg	Right	
	And you can even do like that:	C25	Disability	Leg	Left	
	You say a comamnd in different iterations:	C26	Disability	Seizure	No	
	Iteration 1: **Disability**	C27	Disability	Seizure	No	
	Iteration 2: **Speech**	C29	Disability	Normal	/	
	Iteration 3: **Yes**	C30	Disability	Abnormal	/	
Step 7	Say **Finished** to end the test	C28	Finished	/	/	

Table A.3: The Test Instrument

	Name of Flags	=	String Defination
1	flag_RecordNothing	=	"RecordNothing"
2	flag_RecognizeUnsuccessfully	=	"RecognizeUnsuccessfully"
3	flag_RecognizeFinished	=	"RecognizeFinished"
4	flag_RecognizeABCESuccessfully	=	"RecognizeABCESuccessfully"
5	flag_RecognizeDSuccessfully	=	"RecognizedDSuccessfully"
6	flag_RecognizeGotoStart	=	"RecognizeGotoStart"
7	flag_RecognizeGotoSelection	=	"RecognizeGotoSelection"
8	flag_RecognizeGotoRegister	=	"RecognizeGotoRegister"
9	flag_RecognizeGotoMode	=	"RecognizedGotoMode"
10	flag_RecognizeTwoCommand_Mistake	=	"RecognizeTwoCommand_Mistake"
11	flag_RecognizeOneCommand_Mistake	=	"RecognizeOnecommand_Mistake"
12	flag_Recognize_Single_ABCE_Bit0	=	"Recognize_ABCE_Bit0"
13	flag_Recognize_Single_ABCE_Bit1	=	"Recognize_Single_ABCE_Bit1"
14	flag_Recognize_Single_D_Bit0	=	"Recognize_Single_D_Bit0"
15	flag_Recognize_Single_D_Bit1	=	"Recognize_Single_D_Bit1"
16	flag_Recognize_Single_D_Bit2	=	"Recognize_Single_D_Bit2"

Table A.4: Operation Flag Defination

Test Data, Case1

Tester Tester0 **Place** A quiet room **Filter** No **Noise value** 17dB

General Info.		ABCDE Assessment														
		numbers of (times)						(seconds)		(%)	%(Min)	%(Max)	(seconds)	(Min)	(Max)	(Ave.)
Date	Start Time	Iteration	Recorded Nothing	Unsuc. Recog.	Do Not Match User Mistake	Succ. Recog.	Total words	Duration time	Average time for 1 word	Iteration Accuracy	Recognition Accuracy		totalRecordingTime	Recording Time(seconds)		
2021-07-21	14:41:46	38	1	4	0	33	55	343	3.12	87%	79%	92%	85	1	5	2.30
2021-07-22	10:49:34	35	0	3	0	32	60	291	2.43	91%	87%	95%	101	2	4	2.89
2021-07-22	10:56:39	37	0	6	0	31	59	298	2.53	84%	77%	91%	113	2	5	3.05
2021-07-22	11:03:26	35	0	2	0	33	60	301	2.51	94%	91%	97%	111	2	5	3.17
2021-07-22	11:15:02	35	0	2	0	33	59	241	2.04	94%	91%	97%	69	1	3	1.97
2021-07-22	11:25:35	40	0	4	0	36	61	302	2.48	90%	84%	94%	99	1	6	2.48
2021-07-22	16:48:03	44	0	7	0	37	65	312	2.40	84%	76%	90%	104	1	5	2.36
2021-07-22	18:28:28	36	0	3	0	33	60	266	2.22	92%	87%	95%	96	1	5	2.67
2021-07-22	18:34:48	34	0	4	0	30	57	251	2.20	88%	83%	93%	91	1	6	2.68
2021-07-23	07:25:52	35	0	5	0	30	60	329	2.74	86%	80%	92%	105	2	5	3.00
Case1	**Ave.**	39.9	0.1	4	0	32.8	59.6	293.4	2.47	89%	84%	94%	97.4	1.4	4.9	2.66

Table A.5: Case1: General Test, **E1**, without Filter

Test Data, Case2.1

Tester Tester0 **Place** Living room with TV playing **Filter** No **Noise value** 38dB

General Info.		ABCDE Assessment														
		numbers of (times)						(seconds)		(%)	%(Min)	%(Max)	(seconds)	(Min)	(Max)	(Ave.)
Date	Start Time	Iteration	Recorded Nothing	Unsuc. Recog.	Do Not Match User Mistake	Succ. Recog.	Total words	Duration time	Average time for 1 word	Iteration Accuracy	Recognition Accuracy		totalRecordingTime	Recording Time(seconds)		
2021-07-18	11:51:53	34	0	3	0	31	59	264	2.24	91%	87%	95%	99	1	7	2.91
2021-07-18	12:08:01	42	0	12	0	30	62	330	2.66	71%	63%	84%	121	2	6	2.88
2021-07-20	13:41:24	37	0	3	0	34	63	298	2.37	92%	88%	95%	108	1	4	2.92
2021-07-22	12:23:03	39	0	4	0	35	64	294	2.30	90%	84%	94%	110	1	16	2.82
2021-07-22	12:39:06	48	1	10	0	37	61	354	2.90	77%	65%	85%	120	1	8	2.55
2021-07-22	12:46:40	40	0	7	0	33	62	301	2.43	82%	75%	90%	101	1	4	2.52
2021-07-22	13:01:27	39	0	6	0	33	59	281	2.38	85%	77%	91%	97	1	3	2.49
2021-07-22	13:12:17	47	2	7	0	38	62	354	2.86	81%	70%	87%	106	1	5	2.36
2021-07-22	13:52:32	37	0	4	0	33	62	312	2.52	89%	84%	94%	117	2	7	3.16
2021-07-22	14:03:17	41	0	8	1	33	58	303	2.61	80%	71%	88%	111	1	5	2.71
Case2.1	**Ave.**	40.4	0.3	6.4	0.1	33.7	61.2	309.1	2.52	84%	76%	90%	109	1.2	6.5	2.73

Table A.6: Case2.1, General Test, in a living room, with TV program playing, without filter

Test Data, Case2.2

Tester Tester0 **Place** Living room with TV program playing **Filter** Yes **Noise value** 38dB

General Info.		ABCDE Assessment														
		numbers of (times)						(seconds)		(%)	%(Min)	%(Max)	(seconds)	(Min)	(Max)	(Ave.)
Date	Start Time	Iteration	Recorded Nothing	Unsuc. Recog.	Do Not Match User Mistake	Succ. Recog.	Total words	Duration time	Average time for 1 word	Iteration Accuracy	Recognition Accuracy		totalRecordingTime	Recording Time(seconds)		
2021-07-20	08:18:00	36	1	3	0	32	62	299	2.41	89%	87%	95%	111	1	6	3.17
2021-07-20	20:52:32	35	0	3	0	32	59	245	2.08	91%	87%	95%	82	1	4	2.34
2021-07-20	20:59:39	35	0	2	0	33	59	252	2.14	94%	91%	97%	81	1	3	2.31
2021-07-20	21:07:21	36	0	2	0	34	61	262	2.15	94%	91%	97%	82	1	4	2.28
2021-07-20	21:14:23	34	0	2	0	32	59	251	2.13	94%	91%	97%	90	2	4	2.65
2021-07-20	21:22:58	38	0	8	0	30	60	315	2.63	79%	71%	88%	106	2	5	2.79
2021-07-20	21:30:20	36	0	3	0	33	60	300	2.50	92%	87%	95%	112	1	6	3.11
2021-07-20	21:49:37	36	0	4	0	32	56	308	2.75	89%	82%	93%	116	2	5	3.22
2021-07-21	08:13:05	36	1	5	0	30	61	250	2.05	83%	77%	91%	89	2	4	2.54
2021-07-21	08:20:36	37	0	3	0	34	60	261	2.18	92%	87%	95%	87	1	5	2.35
Case2.2	**Ave.**	35.9	0.2	3.5	0	32.2	59.7	274.3	2.30	90%	85%	94%	95.6	1.4	4.6	2.68

Table A.7: Case2.2, General Test in the living room with TV program playing, with the filter on

Table A.8: Case3, General Test, in the center of a small town Gråbo

Test Data, Case3

Tester Tester0 **Place** Gråbo center **Filter** No **Noise value** 63dB

General Info.		ABCDE Assessment															
		numbers of (times)						(seconds)		(%)	%(Min)	%(Max)	(seconds)	(Min)	(Max)	(Ave.)	
Date	Start Time	Iteration	Recorded Nothing	Unsuc. Recog.	Do Not Match User Mistake	Succ. Recog.	Total words	Duration time	Average time for 1 word	Iteration Accuracy	Recognition Accuracy		totalRecordingTime	Recording Time(seconds)			
2021-07-23	16:13:10	42	3	5	0	34	59	389	3.30	81%	71%	88%	121	1	5	3.1	
2021-07-23	16:35:27	48	3	5	0	40	60	388	3.24	83%	71%	88%	104	1	9	2.31	
2021-07-23	16:45:16	50	6	2	0	42	61	429	3.52	84%	72%	88%	122	1	12	2.77	
2021-07-23	16:55:37	39	0	3	0	36	60	357	2.98	92%	87%	95%	125	1	5	3.21	
2021-07-23	17:09:13	28	2	1	0	25	49	251	2.56	89%	84%	94%	75	1	5	2.88	
Case3	**Ave.**	41.4	2.8	3.2	0	35.4	57.8	362.8	3.12	86%	77%	91%	109.4	1	7.2	2.85	

Test Data, Case4

General Info						ABCDE Assessment												
						numbers of (times)				(seconds)		(%)	%(Min)	%(Max)	(seconds)	(Min)	(Max)	(Ave.)
Tester	Gender	English Level	Accent	Place	NoiseFilter	Iteration	Recorded Nothing	Succ. Recog.	Total words	Duration time	Average time for 1 word	Iteration Accuracy	Recognition Accuracy		totalRecordingTime	Recording Time(seconds)		
Tester1	female	5	No	Distance	SANT	121	64	45	56	1308	11.68	37%	20%	42%	185	2	6	3.25
Tester2	male	5	No	Distance	SANT	115	18	48	74	1118	7.56	42%	27%	52%	401	1	7	4.13
Tester3	male	5	No	Garden	SANT	29	0	28	60	219	1.83	97%	95%	98%	74	1	4	2.55
Tester4	male	5	No	Garden with noisy trafic	SANT	27	0	27	59	290	2.46	100%	100%	100%	92	1	5	3.41
Tester5	male	4	Yes	Quiet room	SANT	42	5	30	57	448	3.93	71%	61%	83%	134	1	7	3.62
Tester6	female	4	Yes	Garden	SANT	43	2	34	63	426	3.38	79%	70%	88%	115	1	7	2.80
Tester7	female	4	Yes	Livingroom with TV on	SANT	34	1	29	57	328	2.88	85%	79%	92%	125	1	10	3.79
Tester8	male	5	No	Livingroom with TV on	SANT	29	0	28	61	301	2.47	97%	95%	98%	119	1	6	4.10

Table A.9: Case4, Public testing, Results of the 8 testers

Table A.10: Test result of tester2 in midterms evaluation

Talk2Me

Testing Analysing

Name of Tester:	Tester2	**Testing Date:**	20210429
Mother Language:	Mandarin	**English Level:**	5
Testing times:	The 7th time	**Grade**	5
Testing Environment:	In the grouproom of Chalmers		

No	Item	Details	Result
1	Speech Recognition	Can Talk2Me hear you?	Yes
2		Most of times, Talk2Me understand you good?	Yes
3		Approximante Accuracy Rate	90-95%
4		The maximal times for you to be 'Understand' by	1
5		Are you satisfied with the response time from Talk2Me?	Yes
1	Speech Synthesis	Are you satisfied with the response time from	Yes
2		Do you think that it is good enought to be used in an Application?	Acceptable
1	Prototype Structure	Are you satisfied?	Yes
2		Do you like the style that Talk2Me talk to you?	Yes
3		Do you think that it is necessary that you need make	Yes
4		Have you finished one/several ABCDE assessment sucessfully?	Yes

Testing Result Problems Analysing
1) After the tester is familiar with the software, the tester made a perfect test. Almost no mistake! 2) Only one mistake: He said Male, but it was recognized as mail.

Possible Solution
1) Male or main, the pronouciation is the same. This problem can be solved easily. Optimize the code.

B

List of Graphs

B.1 GUI of Talk2Me

Figure B.1: The first page of GUI for Talk2Me

B.2 Testing site at Gråbo center

Figure B.2: Test site Gråbo center, E3, average noise value in 10 minute: 63dB

References

[1] K. Forslund, "Challenges in prehospital emergency care", 2007, ISSN: 1652-1153. (Accessed on: 2021).

[2] World Health Organisation, "Pre-hospital trauma care systems", WTO, Tech. Rep. 2, 2005, pp. 191–195.

[3] LidiaMartínez-Sanchez, DanielMartínez-Milln, and V. Ferrés-Padró, "Prehospital emergency care of patients exposed to poisoning: Assessment of epidemiological, clinical characteristics and quality of care", pp. 37–45, 2020.

[4] T. Thim *et al.*, "Initial assessment and treatment with the Airway, Breathing, Circulation, Disability, Exposure (ABCDE) approach", vol. 5, pp. 117–121, 2012. DOI: 10.2147/IJGM.S28478.

[5] M. Short and S. Goldstein, "Ems documentation", 2020.

[6] G. Regel *et al.*, "Prehospital care, importance of early intervention on outcome.", *Acta anaesthesiologica Scandinavica. Supplementum*, vol. 110, pp. 71–76, 1997. DOI: 10.1111/j.1399-6576.1997.tb05508.x.

[7] N. Georges Badr, "Could ict be harnessed for prehospital emergency medical services? the case of the lebanese red cross", pp. 269–276, DOI: 10.5220/0005671902690276. (Accessed on: Aug. 30, 2021).

[8] C. L. Noergaard Bech *et al.*, "Patients in prehospital transport to the emergency department: A cohort study of risk factors for 7-day mortality", *European Journal of Emergency Medicine*, vol. 25, no. 5, pp. 341–347, Oct. 2018. DOI: 10.1097/MEJ.0000000000000470.

[9] V. Lindström, K. Bohm, and L. Kurland, "Prehospital care in sweden", *Notfall + Rettungsmedizin*, vol. 18, pp. 107–109, Mar. 2015. DOI: 10.1007/s10049-015-1989-1.

[10] A. Kemppainen and M. Martinsson, "Speech recognition in prehospital care", Chalmers University of Technology, Tech. Rep., 2021.

[11] K. Wibring *et al.*, "Towards definitions of time-sensitive conditions in prehospital care", *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine 2020 28:1*, vol. 28, no. 1, pp. 1–3, Jan. 2020. DOI: 10.1186/S13049-020-0706-3.

[12] L. Simon, "Beyond EMS Data Collection: Envisioning an Information-Driven Future for Emergency Medical Services", Washington, DC: National Highway Traffic Safety Administration, Tech. Rep., 2016.

[13] "The ABCDE Approach", UK, Resuscitation Council, Tech. Rep., 2015.

[14] T. Olgers *et al.*, "The ABCDE primary assessment in the emergency department in medically ill patients: an observational pilot study.", *Netherlands Journal of Medicine (2017) 75(3) 106-111*, 2017, ISSN: 03002977.

[15] R. C. UK, *The ABCDE Approach*. [Online]. Available: `https://www.resus.org.uk/library/abcde-approach` (Accessed on: Jul. 28, 2021).

[16] "The ABCDE and SAMPLE History Approach Basic Emergency Care Course", World Health Organization, Tech. Rep., 2018.

[17] *Widgets — Kivy 2.0.0 documentation*. [Online]. Available: `https://kivy.org/doc/stable/guide/widgets.html` (Accessed on: Jul. 27, 2021).

[18] D. Zhang, J. Wang, and M. Sun, *The Progress That Natural Language ProcessingHas Made Towards Human-level AI*, 2020. DOI: `10.23977/jaip.2020.030107`.

[19] S. R. Joseph *et al.*, "Natural Language Processing: A Review", *International Journal of Research in Engineering and Applied Sciences*, vol. 6, no. 3, pp. 1–8, 2016, ISSN: 2249-3905.

[20] M. A. Hearst, "Mixed-Initiative Systems", *IEEE Intelligent Systems*, vol. 14, no. 5, p. 14, 1999.

[21] J. W. Buck, S. Perugini, and T. V. Nguyen, "Natural language, mixed-initiative personal assistant agents", *ACM International Conference Proceeding Series*, Jan. 2018. DOI: `10.1145/3164541.3164609`.

[22] M. Walker and S. Whittaker, "Mixed Initiative in Dialogue: An Investigation into Discourse Segmentation", University of Pennsylvania and Hewlett Packard Laboratories, Tech. Rep.

[23] "Noise reduction", Tech. Rep., 2004. (Accessed on: Jul. 4, 2021).

[24] Y. Lu and P. C. Loizou, "A geometric approach to spectral subtraction", 2008. DOI: `10.1016/j.specom.2008.01.003`.

[25] N. Upadhyay and A. Karmakar, "Peer-review under responsibility of organizing committee of the eleventh international multi-conference sciencedirect speech enhancement using spectral subtraction-type algorithms: A comparison and simulation study", *Procedia Computer Science*, vol. 54, pp. 574–584, 2015. DOI: `10.1016/j.procs.2015.06.066`.

[26] S. V. Vaseghi, "1 SPECTRAL SUBTRACTION 11.1 Spectral Subtraction 11.2 Processing Distortions 11.3 Non-Linear Spectral Subtraction 11.4 Implementation of Spectral Subtraction", 2000.

[27] A. Becker, *Kalman filter tutorial*, 2018. [Online]. Available: `https://www.kalmanfilter.net/default.aspx`.

[28] K. F. Applications, "Subject MI63: Kalman Filter Tank Filling", Dept. of Instrumentation and Electronics Engineering,Jadavpur University, Tech. Rep., 2008.

[29] O. Das, "Kalman Filter in Speech Enhancement", *IEEE Control Systems*, 2016, ISSN: 2012-2013.

[30] M. S. Grewal and A. P. Andrews, "Applications of Kalman Filtering in Aerospace 1960 to the Present", vol. 30, no. 3, pp. 69–78, 2010. DOI: 10.1109/MCS.2010.936465. [Online]. Available: https://https://ieeexplore.ieee.org/document/5466132.

[31] C. Torsten Wik, *Discrete time kalman filter*, 2020.

[32] Q. N. Tran and H. R. Arabnia, *Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology*. 2016, ISBN: 978-0-12-804203-8.

[33] *ICSI Speech FAQ - 4.1 How is the SNR of a speech example defined?* (Accessed on: Aug. 1, 2021).

[34] A. K. Martinsson Marcus, "Speech Recognition in Prehospital Care", Tech. Rep., 2020.

[35] "SpeechRecognition · PyPI", 2021. [Online]. Available: https://pypi.org/project/SpeechRecognition/.

[36] E. Grossi and M. Buscema, "Introduction to artificial neural networks", *European Journal of Gastroenterology and Hepatology*, vol. 19, no. 12, pp. 1046–1054, 2007, ISSN: 0954691X. DOI: 10.1097/MEG.0b013e3282f198a0.

[37] D. Jurafsky and H. M. James, "Neural Networks and Neural Language Models", *Speech and Language Processing*, no. Chapter 9, 2020.

[38] S. D. M. Jane Jaleel Stephan and M. K. Abbas, "Neural Network Approach to Web Application Protection", International Journal of Information and Education Technology, Tech. Rep., 2015.

[39] T. Szandała, "Review and comparison of commonly used activation functions for deep neural networks", Oct. 2020.

[40] J. Feng and S. Lu, "Performance Analysis of Various Activation Functions in Artificial Neural Networks", *Journal of Physics: Conference Series*, vol. 1237, no. 2, 2019, ISSN: 17426596. DOI: 10.1088/1742-6596/1237/2/022030.

[41] K. Nantomah, "On some properties of the sigmoid function", vol. 3, pp. 79–90, Apr. 2019.

[42] I. C. Education, *What are Neural Networks?*, 2020. [Online]. Available: https://www.ibm.com/cloud/learn/neural-networks (Accessed on: Jul. 28, 2021).

[43] *Real-Life Applications of Neural Networks | Smartsheet*. [Online]. Available: https://www.smartsheet.com/neural-network-applications (Accessed on: Jul. 30, 2021).

[44] F. Bre, J. M. Gimenez, and V. D. Fachinotti, "Prediction of wind pressure coefficients on building surfaces using artificial neural networks", *Energy and Buildings*, vol. 158, no. November, pp. 1429–1441, 2018. DOI: 10.1016/j.enbuild.2017.11.045.

[45] K. Shiruru, "An introduction to artificial neural network", *International Journal of Advance Research and Innovative Ideas in Education*, vol. 1, pp. 27–30, Sep. 2016.

[46] I. GAUTAM, "Speech synthesis", Manav Rachna International University, Tech. Rep., 2011.

[47] T. Dutoit, "High-quality text-to-speech synthesis: An overview", Tech. Rep., 1997, pp. 25–36.

[48] I. Gautam, "Speech Synthesis Technology", pp. 1–2, 2005.

[49] J. O. Onaolapo and F. E. Idachaba, "A simplified overview of text-to-speech synthesis", Covenant University Repository, Tech. Rep., 2014.

[50] National Association of Emergency Medical Technicians NAEMT, Ed., *PHTLS: prehospital trauma life support*, Ninth edition, Burlington, Massachusetts: Jones & Bartlett Learning, 2020, 762 pp., ISBN: 978-1-284-17147-1.

[51] *Comparing Python to Other Languages | Python.org*, 2021. [Online]. Available: `https://www.python.org/doc/essays/comparisons/` (Accessed on: May 6, 2021).

[52] M. Leuthäuser, "Object-oriented programming with Python An introduction into OOP and design patterns with Python", Tech. Rep.

[53] *Numpy*, 2021. [Online]. Available: `https://numpy.org/`.

[54] *Matplotlib: Visualization with Python*, 2021. [Online]. Available: `https://matplotlib.org/`.

[55] *Introduction — Kivy 2.0.0 documentation*, 2021. [Online]. Available: `https://kivy.org/doc/stable/gettingstarted/intro.html` (Accessed on: May 6, 2021).

[56] *Features - PyCharm*. [Online]. Available: `https://www.jetbrains.com/pycharm/features/` (Accessed on: Sep. 4, 2021).

[57] *Speech Recognition in Real-Life Background Noise by Young and Middle-Aged Adults with Normal Hearing*. DOI: `C2015-0-01779-8`. (Accessed on: Aug. 1, 2021).

[58] *pyttsx3 · PyPI*, 2020. [Online]. Available: `https://pypi.org/project/pyttsx3/` (Accessed on: May 7, 2021).

[59] *Plexgear SV-120 Datorheadset – Plexgear*. [Online]. Available: `https://www.sommarkyla.se/shop/plexgear-sv120-datorheadset` (Accessed on: Sep. 5, 2021).

[60] *EXAMOBILE | Software house*. [Online]. Available: `https://examobile.pl/en/home-en/` (Accessed on: Sep. 5, 2021).

[61] *APPLICABILITY | meaning in the Cambridge English Dictionary*. (Accessed on: Sep. 6, 2021).

[62] "Artificial Intelligence and privacy", *Social Science Computer Review*, 2018, ISSN: 0894-4393.

[63] C. Bartnech and C. Lutge, "Privacy Issues of AI", Tech. Rep., 2021, pp. 61–70. DOI: `10.1007/978-3-030-51110-4_8`.

[64] C. Ittichaichareon, "Speech recognition using MFCC", *Conference on Computer*, pp. 135–138, 2012, ISSN: 17356865. arXiv: `arXiv:1011.1669v3`.