

Master Thesis



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Measurement

Digital signal preprocessing for content recognition improvement

Aytaj Sabitova

Supervisor: prof. Ing. Jan Holub, Ph.D.
Field of study: Cybernetics and Robotics
May 2024

I. Personal and study details

Student's name: **Sabitova Aytaj** Personal ID number: **516477**
Faculty / Institute: **Faculty of Electrical Engineering**
Department / Institute: **Department of Measurement**
Study program: **Cybernetics and Robotics**

II. Master's thesis details

Master's thesis title in English:

Digital signal preprocessing for content recognition improvement

Master's thesis title in Czech:

P edzpracování digitálního signálu pro zlepšení rozpoznávání obsahu

Guidelines:

Based on overview of state-of-the art methods for recorded audio signal enhancement to increase the success rate of speech radio channel content recognition, design and characterise the selected procedures achieving the best results. Student can deploy and compile together existing tools or develop own libraries or functions, however, the detailed description of the functionality is essential. DNN or other AI processes is welcome but not mandatory. Create training and testing speech recordings databases and demonstrate the results of the procedure.

Bibliography / sources:

- [1] Jinyu Li, Li Deng, Reinhold Häb-Umbach, Yifan Gong: Robust Automatic Speech Recognition, ISBN: 9780128023983
- [2] Demir, C., Dogan, M U., Cemgil, A T., & Saraclar, M. (2012, April 1). Catalog-based single-channel speech-music separation for automatic speech recognition.
- [3] Ma, D. (2021, January 1). Multitask-Based Joint Learning Approach To Robust ASR For Radio Communication Speech.
- [4] Andrew Catellier and Stephen Voran: Wideband Audio Waveform Evaluation Networks: Efficient, Accurate Estimation of Speech Qualities

Name and workplace of master's thesis supervisor:

prof. Ing. Jan Holub, Ph.D. Department of Measurement FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **15.02.2024** Deadline for master's thesis submission: **24.05.2024**

Assignment valid until:

by the end of summer semester 2024/2025

prof. Ing. Jan Holub, Ph.D.
Supervisor's signature

Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature

Acknowledgements

I would like to convey my sincere gratitude to my supervisor, prof. Ing. Jan Holub, Ph.D., for introducing me to the field of digital signal processing and automatic speech recognition. He has been a constant source of encouragement and insight during my research and helped me with numerous problems and professional advancements. His willingness to generously give his time for immediate consultations and suggestions during this research, whenever I was in need, has been very much appreciated. I would like to thank my family and friends for their continuous support throughout my studies.

Declaration

I declare that the presented thesis is all my work, and I have cited all sources of information following the methodical guidelines for ethical principles in the preparation of an academic final thesis.

Prague, May 24, 2024

Prohlašuji, že předložená práce je výsledkem mé vlastní práce a všechny zdroje informací jsem citoval/a v souladu s metodickými pokyny pro etické zásady při přípravě akademické závěrečné práce.

V Praze, 24 května 2024

Abstract

Speech is the main way people communicate with each other, and we cannot even imagine our lives without it. Since a century ago, there have been lots of works dedicated to speech technologies. Automatic Speech Recognition (ASR), is one of the important bridges between human-machine interactions. Working with this technology is always considered important.

In the last decades, ASR has seen enormous improvements, making significant improvements in several key areas. Thanks to these improvements, ASR has become not only a part of our daily lives but also an important system in diverse areas such as military communications, air traffic control, public safety networks, and emergency response systems. Despite these improvements, ASR still lacks realistic situations and shows performance drops in some aspects. There is evidence that these systems show different performances between genders, and some languages are less favorable for ASR compared to others.

In this thesis, we focus on increasing the success rate of speech radio channel content recognition. We aim to design and characterize procedures that achieve the best results. Our research can be divided into main areas: first, we provide an overview of state-of-the-art methods on the topic; then, we present our approach and simulated method that we intend to use; finally, we present and compare results for discussion. In conclusion, this research provides a comprehensive understanding of speech, digital signal processing, and Automatic Speech Recognition (ASR) systems, offering valuable insights into the optimization of these systems.

Keywords: Speech, Automatic Speech Recognition, Digital Signal Processing, Word Error Rate, WER, Audio and speech processing, Speech-to-text.

Supervisor: prof. Ing. Jan Holub,
Ph.D.
Fakulta elektrotechnická,
Technická 1902/2,
Prague 6,
Czech Republic

Abstrakt

Mluvení je hlavní způsob, jak lidé komunikují mezi sebou, bez kterého si téměř neumíme představit své životy. Již více jak století je věnováno mnoho pozornosti technologiím pro záznam a zpracování hlasu. Automatické rozpoznávání řeči (ASR) je jedním z důležitých prvků interakce člověka a stroje. Výzkum v této oblasti technologie je v současné době velmi aktuální.

V posledních desetiletích ASR zaznamenalo velké zlepšení, které přineslo významné pokroky v několika klíčových oblastech. Díky těmto vylepšením se ASR stal nejen součástí našich každodenních životů, ale také důležitým systémem v mnoha oblastech, jako jsou vojenské komunikace, řízení leteckého provozu, veřejné bezpečnostní sítě a systémy pro nouzovou reakci. Přesto je ASR stále nedokonalé v realistických situacích, kdy vykazuje pokles výkonu v některých aspektech. Je prokázáno, že tyto systémy vykazují různé výkonnosti mezi pohlavími a také některé jazyky jsou méně příznivé pro ASR ve srovnání s jinými.

V této práci se zaměřujeme na zvýšení úspěšnosti rozpoznávání obsahu radiového kanálu. Cílem je navrhnout a charakterizovat postupy, které dosahují nejlepších výsledků. Náš výzkum lze rozdělit do těchto hlavních oblastí: nejprve poskytujeme přehled nejnovějších metod v dané oblasti; poté představujeme náš přístup a simulovanou metodu, kterou plánujeme použít; nakonec představujeme a porovnáváme výsledky k diskusi. Celkově tato práce poskytuje ucelený přehled hlasových číslicově zpracovaných signálů a systémů automatického rozpoznávání řeči (ASR), a nabízí cenné poznatky pro optimalizaci těchto systémů.

Klíčová slova: Speech, Automatic Speech Recognition, Digital Signal Processing, Word Error Rate, WER, Audio and speech processing, Speech-to-text.

Překlad názvu: Předzpracování digitálního signálu pro zlepšení rozpoznávání obsahu

Contents

1 Introduction	1	5.1.4 Audacity x RX 10.....	39
1.1 General overview	1	5.2 Speech-to-Text tools	40
1.2 Thesis objectives and organization	2	5.2.1 Whisper	40
2 Speech	5	5.2.2 Vosk	43
2.1 Formation of speech signal	5	6 Simulated experiments	45
2.2 Physiology of hearing	7	6.1 Methodolgy	45
2.3 Properties of speech signal	8	6.2 Presentation of experimental data	46
2.4 Common speech signal descriptors	9	6.3 Test setup	47
2.4.1 Fundamental frequency	9	6.4 Calculation of ASR results	49
2.4.2 Sound intensity and loudness	10	6.5 Ethical Considerations guidelines	
2.4.3 Spectral characteristics	12	for ASR implementation	50
2.5 Phonetic Differences Between Male		7 Practical experiment	51
and Female Speech	12	7.1 Guidelines for implementing ASR	
3 Digital Signal Processing (DSP)	15	in environments	51
3.1 Introduction of Digital Signal		7.2 Experimental Results	54
Processing(DSP).....	15	7.2.1 First Experiment	54
3.2 Digital Signal Processing Basic		7.2.2 Second Experiment.....	56
Theory	16	7.2.3 Third Experiment.....	59
3.3 Digital Signal Processing systems	17	7.2.4 Fourth Experiment	62
3.4 Advantages and Disadvantages of		7.3 Analysis of ASR system	
digital signal processing technology	18	performance and key findings.....	65
4 Automatic Speech Recognition	21	7.4 Future Research Directions in	
4.1 Introduction of ASR	21	ASR	67
4.1.1 Speech Signal.....	22	8 Conclusion	69
4.1.2 Speech Coding.....	23	Bibliography	73
4.1.3 Text-to-Speech Synthesis....	24	A Abbreviations list	79
4.2 Feature Extraction.....	25		
4.2.1 Mel frequency cepstral			
coefficients (MFCC)	26		
4.2.2 Linear prediction coefficients			
(LPC)	28		
4.2.3 Linear prediction cepstral			
coefficients (LPCC)	29		
4.2.4 Line spectral frequencies (LSF)	30		
4.2.5 Discrete wavelet transform			
(dwt)	31		
4.2.6 Perceptual linear prediction			
(PLP)	32		
4.3 Error rate analysis of current ASR			
systems.....	34		
5 Software tools	35		
5.1 Background noise-cleaning tools	35		
5.1.1 Descript	35		
5.1.2 iZotope RX 10.....	36		
5.1.3 Audacity.....	38		

Figures

2.1 Schematic view of human speech production mechanism.[1]	6	6.2 Code to calculate Word Error Rate (WER)	50
2.2 Block diagram of human speech production system.[1]	6	7.1 Descript.	51
2.3 The anatomy of the human ear.[2]	7	7.2 Descript.	52
2.4 The ear scheme.[3]	8	7.3 Whisper AI(local).	52
2.5 Fletcher-Munson's curves of equivalent perceived loudness.[4]	11	7.4 Whisper AI (cloud) code.	53
3.1 A digital processing scheme.[5]	15	7.5 Vosk code	54
3.2 Digital Signal Processing Flow.[6]	16	7.6 Comparison chart of Descript.	55
4.1 The Speech Chain: from message to the speech signal to understanding.[7]	23	7.7 Comparison WER scores of Descript.	56
4.2 Speech coding block diagram — encoder and decoder.[7]	23	7.8 Comparison chart of Whisper(local).	58
4.3 Text-to-speech synthesis system block diagram.[7]	24	7.9 Comparison WER scores of Whisper(local).	59
4.4 Block diagram of MFCC processor.[8]	27	7.10 Comparison chart of Whisper(cloud).	61
4.5 Block diagram of LPC processor.[8]	29	7.11 Comparison WER scores of Whisper(cloud).	62
4.6 Block diagram of LPCC processor.[8]	30	7.12 Comparison chart of Vosk.	64
4.7 Block diagram of LSF processor.[8]	31	7.13 Comparison WER scores of Vosk.	65
4.8 Block diagram of DWT.[8]	32		
4.9 Block diagram of PLP processor.[8]	33		
4.10 WER calculations for the "I love you" example.[9]	34		
5.1 Descript.	36		
5.2 iZotope RX 10	37		
5.3 iZotope RX 10	37		
5.4 iZotope RX 10	38		
5.5 Audacity	38		
5.6 Audacity	39		
5.7 Audacity x RX 10	40		
5.8 Audacity x RX 10	40		
5.9 Whisper AI(local)	42		
5.10 Whisper AI(cloud)	43		
5.11 Vosk	43		
6.1 Our approach	45		

Tables

6.1 Male speech database	46
6.2 Female speech database	47
6.3 Audio tools	47
6.4 ASR tools	48
6.5 First Experiment Setup.....	48
6.6 Second Experiment Setup.....	48
6.7 Third Experiment Setup.....	48
6.8 Fourth Experiment Setup.....	49
7.1 First Experiment Setup.....	55
7.2 WER results for the first experiment.....	55
7.3 WER results for the first experiment.....	56
7.4 Second Experiment Setup.....	57
7.5 WER of Whisper AI(local) x raw data	57
7.6 WER results for the second experiment.....	57
7.7 WER results for the second experiment.....	58
7.8 Third Experiment Setup.....	59
7.9 WER of Whisper AI(cloud) x raw data	60
7.10 WER results for the third experiment.....	60
7.11 WER results for the third experiment.....	61
7.12 Fourth Experiment Setup	62
7.13 WER of Vosk x raw data	63
7.14 WER results for the fourth experiment.....	63
7.15 WER results for the fourth experiment.....	64

Chapter 1

Introduction

1.1 General overview

Speech has always been the main way of human communication. We cannot imagine our daily lives without speaking to each other; it is the most natural interaction between humans. In different kinds of real-life situations, it's the fastest way for us to communicate. In today's rapidly improved technology and AI, there are several works in which machines are replacing us, humans. Because of these changes, human interaction is not limited to just other humans, but also they are interacting with machines. Day by day, it is becoming more common to see human-machine interaction in different areas of life. Even though technology improved, making several key improvements, there are still lots of real-life scenarios where the performance of systems drops. In order, to make this human-machine interaction more smooth and accurate, research continues to find ways and improve this communication as much as possible.

When it comes to Automatic Speech Recognition (ASR) systems, there are two different purposes. Firstly, they are the backbone of the speech-to-text (STT) systems. These systems are usually used for transcribing related works. This means it used longer parts of human speech in texts in systems. For impaired people, it can be used as an automatic subtitle generation systems e.g. real-time situations. Secondly, these systems help us to control electronic devices which means we can enable electronic devices by our speech in the voice-control domain. These systems are the text-to-speech (TTS) systems. These systems transform the written text into sound which means they are allowing computers to "speak". Both systems are usually integrated together as basic modules of higher-level dialogue systems.

The accuracy ASR system plays a crucial role in successful speech transcription or human-machine interaction. Over the years, the structure of ASR systems has changed and improved to more stabilized versions. These versions are most suitable to model individuals as parts of human speech have been found and incorporated into the working whole. The speech signal parametrization is one of the standardized aspects of the overall ASR system flow. Nowadays, most of the parametrizations that are used, are aiming to minimize the differences between speakers and their speaking styles to create

more general ASR acoustic models. It is believed that, like this, whole ASR acoustic models can be very general. It will help to make speech recognition process more robust. However, we can still find various transcription errors in ASR systems in real life. The reason behind those ASR errors depends on the insufficient amount of exact or similar examples in training data to speaker-specific or non-standard realizations of training data. It also shows untypical prosodic properties like intonation, loudness, and speech rate of speech, compared to typical training data, increase the chance of words being misrecognized. This explains the effort to disregard those prosodic properties (mainly fundamental frequency which contributes to perceived pitch) during the signal parametrization. Nevertheless, it can still be claimed that prosody information is a useful cue for ASR systems. For enhancing its output with punctuation marks or decreasing its word error rates (WER) and increasing its accuracy.

1.2 Thesis objectives and organization

The goal of this thesis lies in contributing to a deeper understanding of the role ASR technology plays and creating an experiment and simulations for testing speech recordings. The thesis is organized as follows:

- **Chapter 2** is dedicated to speech and its formation. Through the subchapters, we aimed to present information about speech and speech signals.
- **Chapter 3** is dedicated to Digital Signal Processing and its technology. Through the subchapters, we aimed to present information on Digital Signal Processing and how it has a huge impact on our daily lives.
- **Chapter 4** is dedicated to Automatic Speech recognition(ASR). Through the subchapters, we introduce the main application areas of ASR systems, describe their basic architecture, and then error rate analysis of current ASR systems.
- **Chapter 5** is dedicated to Software tools that are used in this research. Through the following subchapters, we presented information about these tools and their distinct roles and capabilities.
- **Chapter 6** is dedicated to the Simulated experiments in this research. Through the following subchapters, we presented our approach, our data, our test setup, and the ethical considerations for our experiments.
- **Chapter 7** is dedicated to the Practical experiments and their results in this research. Through the following subchapters, we presented how to implement ASR in different environments, and the results after experiments and discussion about key findings. Chapters end with suggestions about future research directions in ASR.

- **Chapter 8** is Conclusion. In the last chapter, we will conclude our research with findings and suggestions for future works.

Chapter 2

Speech

Speech is vocal communication that humans use to express language. When discussing languages, it's remember that each language uses words that are phonetic combinations of vowel and consonant sounds. First humans used to communicate with more basic sounds, kind of similar to the way that animals communicate with each other. However, unlike animals, over the years and through evolution, human speech developed more and become the main communication way for humans. Today, humans perform many different speech acts apart from basic sounds example, informing, answering, directing, and in among other speaking. Through the following subchapters, we will try to present information about speech and speech signals.

2.1 Formation of speech signal

The production of speech signals is a complex physical process. The sound, we hear is created by lungs, glottis (with vocal cords), and articulation tract (mouth and nose cavity). In speaking, the speaker produces a speech signal in the form of pressure waves that travel from the speaker's head to the listener's ears. The creation of speech continues as after the speaker breathes in, causing the muscles in the chest to tighten. Then air is pushed from the lungs against upward gravity into the larynx. The vocal cords are inside the larynx and this air passes through it. If they are vibrating, it is possible to get some periodic natural sound, otherwise, they pass by as noise-characterized sound without any fundamental frequency. This signal is nonstationary changing characteristics as the muscles of the vocal tract contract and relax. For each sound, there is a positioning for each of the vocal tract articulators: vocal cords, tongue, lips, teeth, velum, and jaw. Sounds are typically divided into two broad classes: vowels, which allow unrestricted airflow in the vocal tract; and consonants, which restrict airflow at some point and have a weaker intensity than vowels.

The human speech production system is illustrated in the following picture.

2. Speech

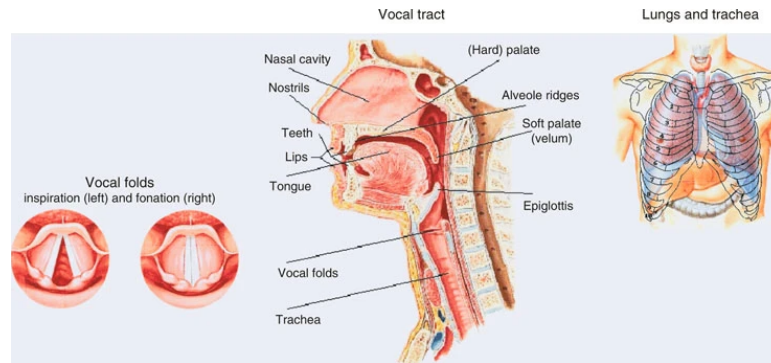


Figure 2.1: Schematic view of human speech production mechanism.[1]

The creation of speech process can be seen as a filtering operation where vocal and nasal tracts, the main cavities of the creation of speech act as acoustic filters. The organs are characterizing the filter and it's loaded at the main output which is radiation impedance because of lips. They are used to change the properties of the filter system and loading over time. The source that excites the filter may be either periodic, resulting in voiced speech, or noisy and aperiodic, resulting in unvoicing speech.

Speech Production model is illustrated in the following picture.

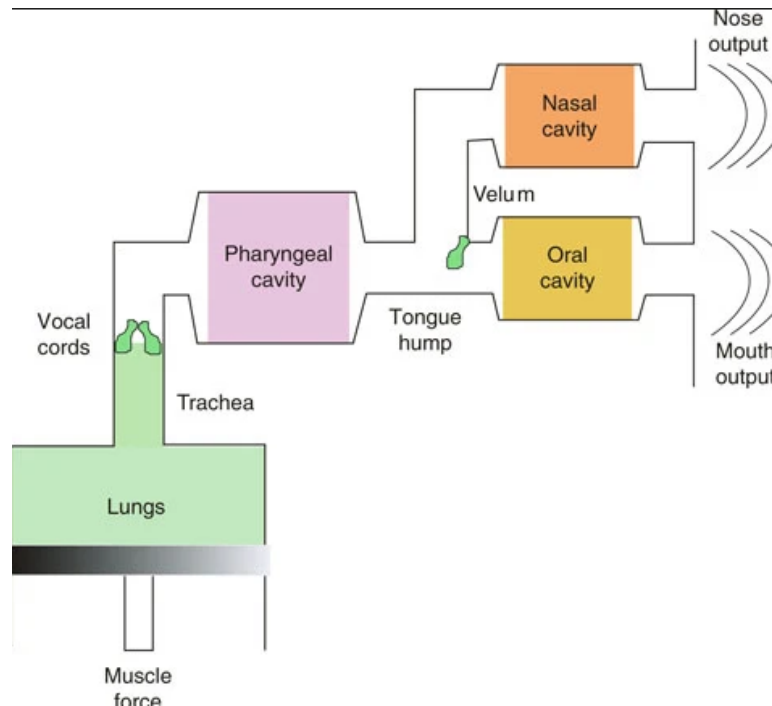


Figure 2.2: Block diagram of human speech production system.[1]

We can assume that the model that the signal produced at the glottal level acts as the source and is linearly filtered by the vocal tract. The sound produced is then emitted into the surrounding air through radiation loading,

typically from the lips. Additionally, the model assumes that the source and filter are independent of each other.[1]

2.2 Physiology of hearing

The human hearing mechanism of ears is based on their location and the form of signal inside it (which can be acoustic, vibration, or electric) can be divided into 3 main parts: the outer ear, middle ear, and inner ear.

The anatomy of the human ear is illustrated in the following picture.

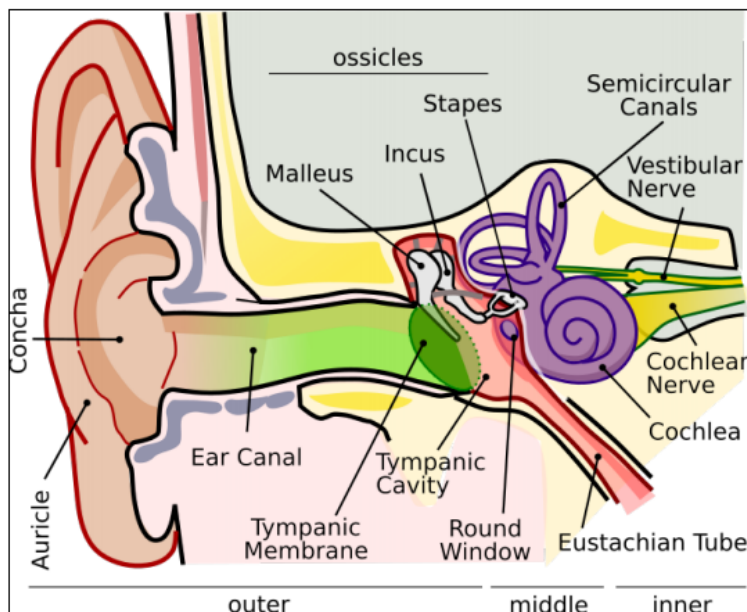


Figure 2.3: The anatomy of the human ear.[2]

Sound initially enters the outer ear, then goes through the middle ear before reaching the inner ear, where it is converted into neural signals that are sent to the brain to create perception. The pinna, the only visible part of the outer ear, primarily influences the positional aspects of hearing for frequencies higher than 500 Hz. Continuing from the outer ear, the tube-shaped ear canal (meatus) maintains a resonating mode, typically around 3 kHz, which is crucial for the intelligibility of speech signals due to its resonance peak of approximately 15 dB. The eardrum (tympanum) separates the outer ear from the middle ear, housing the three ear bones—hammer (malleus), anvil (incus), and stirrup (stapes)—which serve as the center of an effective acoustic conversion system, adapting impedance between the air and fluid environments of the inner and outer ear. Also in the middle ear, there are mechanisms to protect hearing from high-intensity sounds when necessary. The border between the middle and inner ear is marked by oval and round windows. The main organ in the inner ear is the spiral-shaped cochlea, with its tube divided into two floors by the divider containing the

sense organ of Corti. This organ contains hair cells that transmit information to the brain via approximately 30,000 threads of the auditory nerve, using low-voltage electric impulses. Different frequencies stimulate different parts of the hair cells, enabling the brain to perceive the specific frequency content of the stimulus.[2] [3]



Figure 2.4: The ear scheme.[3]

2.3 Properties of speech signal

Speech signals interplay of various properties that reflect both the complexity of human communication and the physiological processes in speech production. While numerous speech sounds can be produced, the shape of the vocal tract and its excitation pattern change relatively slowly over time. As a result, speech can be considered quasi-stationary over short periods (typically around 20-25 milliseconds).

Voiced segments of the speech signal can be defined physiologically as complex tones, which are periodic non-sinusoidal signals composed of multiple frequency components. [10]

Speech signals exhibit noise and variability, stemming from factors such as background noise, speaker characteristics, and environmental conditions. This noise and variability are essential for achieving accurate speech-processing results.

The properties of speech signals are diverse and dynamic which reflects both speech production and human communication. These properties are essential for developing effective speech-processing technologies and gaining insights into the nature of spoken language. These fundamental properties have a significant role in the digital processing of speech signals, particularly in terms of the typical frame size and the applicability of the Fourier transform and other signal transforms. [10]

2.4 Common speech signal descriptors

Typically, signals offer three basic types of descriptors. Pitch information describes the fundamental frequency, particularly in voiced parts of speech. Intensity describes the overall strength of the signal. Spectral descriptors aim to capture the signal's frequency content by highlighting key events in the spectrum, compressing the full information. These descriptors, or their specific subtypes, are commonly used as prosodic features.

2.4.1 Fundamental frequency

Fundamental frequency (f_0) estimation, which is also known as pitch detection, is a very popular research topic for many years, even still today. The basic problem is to extract the fundamental frequency (f_0) from a sound signal, which is usually the lowest frequency component, or partial, which relates well to most of the other partials. In a periodic waveform, most partials are harmonically related, meaning that the frequency of most of the partials is related to the frequency of the lowest partial by a small whole-number ratio. The frequency of this lowest partial is f_0 of the waveform. The (f_0) measure is Hertz [Hz] (number of periods within one second [s^{-1}]).[11]

Fechner-Weber's law outlined that frequency measured in Hertz doesn't directly correlate with human perception of pitch. According to this law, our perception's velocity is proportional to the logarithm of the stimulus. In simpler terms, when the stimulus increases geometrically, our perception rises arithmetically. Therefore, converting from absolute frequency units to relative musical units is essential in tasks aiming to imitate or evaluate human perception accurately.

$$ST_{\text{diff}} = 12 \log_2 \left(\frac{f_2}{f_1} \right) [ST] \quad (2.1)$$

Equation 2.1 establishes a conversion from frequency difference in Hertz to semitone difference (musical scale) using a logarithm based on frequency ratios. This conversion shifts from absolute to relative measurement, requiring a reference point, f_1 , for relativity. To ensure consistency, it's beneficial to relate the musical units to a common frequency unit, such as 1 Hz or 100 Hz. Equation 2.2 defines the conversion into semitones related to 100 Hz [$ST_{\text{rel}100\text{Hz}}$].

$$ST_{\text{rel}100\text{Hz}} = 12 \log_2 \left(\frac{f_2}{100} \right) [ST] \quad (2.2)$$

The unit semitone [st, ST] corresponds to the musical unit in Western music, where an octave (the interval between musically "same" notes) is divided into 12 semitones. For more precise measurement, one can use the semitone cent unit, where one semitone consists of 100 cents.

Several studies have shown that human hearing exhibits a logarithmic nature up to a frequency of around 800 Hz, after which it becomes "non-linear" compared to the logarithm. This discovery led to the suggestion of several alternative scales, such as the mel, bark, and erb scales, which provide even better approximations of fundamental frequency perception.

Two essential descriptors of fundamental frequency period stability, typically assessed during sustained vowel sounds, are jitter and shimmer. Jitter reflects frequency stability by measuring the equality of period durations. It is calculated as the average absolute difference between consecutive periods, divided by the average period duration. Shimmer, conversely, quantifies the amplitude stability of f_0 (fundamental frequency) periods. It is computed as the average absolute difference between the amplitudes of consecutive periods, divided by the average amplitude.

These measures have empirically established thresholds and are primarily employed in speech pathology research. Additionally, the harmonics-to-noise ratio (HNR), which characterizes voice hoarseness, is often utilized. HNR values vary depending on the vowel identity, with excessively low values typically indicating the presence of hoarseness in the voice.[12]

2.4.2 Sound intensity and loudness

Fechner-Weber's law influenced the perception of sound intensity and pitch as well. The intensity of a sound wave refers to the amount of sound energy passing through a unit area per second. Loudness, on the other hand, measures the ear's response to sound, with amplitude determining how loud a sound is perceived. The intensity of a sound is directly related to its amplitude, which affects how loud it sounds to the ear. Decibels (dB) are used to measure intensity. The intensity, or amount of energy, in sound waves, determines the perceived loudness of the sound. As decibel levels increase, sound waves become more intense, resulting in louder noises.

Both objective measures and subjective quantities exist to describe sound intensity, reflecting both its physical properties and human perception.

Two commonly used objective measures for sound level exist. The first is derived from the definition of sound intensity, representing the true physical meaning as emitted power into a unit area (expressed in W/m^2). The second measure is based on sound pressure, which arises from slight atmospheric air pressure modulations. This principle involves the capturing of signals through the deviation of a microphone diaphragm and is directly related to the captured voltage and audio signal amplitude.

Both measures utilize a 10-based logarithm of the ratio as the core transformation from the original unit into the 'level' measure.

Sound intensity level (SIL) is defined as:

$$L_I = 10 \log \left(\frac{I}{I_0} \right) [dB] \quad (2.3)$$

Sound pressure level (SPL) is defined as:

$$L_p = 20 \log \left(\frac{p}{p_0} \right) [dB] \quad (2.4)$$

It can be assumed that SPL and SIL levels correspond to each other.[13]

The strength inherent in the use of decibel units may not be immediately apparent, as it replaces the usual linear difference in basic units with ratios of the original unit, introducing a new kind of arithmetic and understanding.

The subjective perception of loudness is heavily influenced by sound frequency, as sounds of the same intensity can evoke different perceptions of loudness at various frequencies. To address this, numerous experiments have been conducted to derive curves of equivalent perceived loudness, known as Fletcher-Munson curves[4], by comparing the intensities and frequencies of tested sine waves with those at 1kHz. The perceived loudness level in phons [Ph] at 1kHz directly corresponds to the objective sound pressure level (SPL).

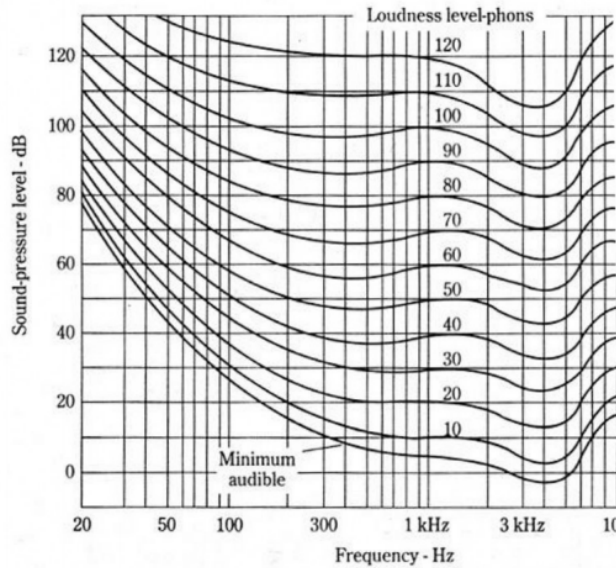


Figure 2.5: Fletcher-Munson's curves of equivalent perceived loudness.[4]

However, it's important to note that these curves are valid for sine signals only. In reality, perceived loudness is affected by various factors beyond sound pressure and frequency, including bandwidth, spectral composition, information content, time structure, and the duration of sound signal exposure.

However, none of the previously presented level measurements accurately reflect the sense of perceived loudness. This discrepancy led to the creation of a new measure called loudness, which is anchored to a loudness level of 40 phons. The following equation defines loudness N in [sone] units as derived from loudness level L [Ph]:

$$N = 2^{\left(\frac{L-40}{10}\right)} [\text{sone}] \quad (2.5)$$

throat, and vocal folds contribute to the sounds produced, individuals also adopt speech patterns associated with their gender identity. This becomes especially evident when comparing the speech of preadolescent boys and girls who have similar anatomical features before puberty.

Beyond pitch, differences in phonation and articulation play significant roles in distinguishing male and female speech.[14]

Let's look at them more closely:

- **Phonation and Pitch:** The fundamental frequency (F0) of the voice, determined by the vibration of the vocal folds, is closely tied to the perceived pitch. Male vocal folds, typically longer and thicker, vibrate at a slower rate, resulting in a lower pitch, with an average F0 of 100–120 Hz in languages like German and English. Meanwhile, female vocal folds, shorter and lighter, vibrate at about twice the frequency, around 200–220 Hz.[14]
- **Articulation Vowel:** Apart from differences in voice pitch, variations in the size and shape of the vocal tract also influence voice quality. The average female vocal tract length is about 14–14.5 cm, while the male vocal tract is 17–18 cm on average, primarily due to the larynx lowering during male puberty. These differences affect sound production, as the vocal tract modifies the tone created at the glottis. Vowel qualities, such as [i], [e], or [o], vary depending on vocal tract configurations and the strengthened frequency components.[14]
- **Articulatory Speed:** Individual measurements of vowels in speech reveal that the articulators are constantly in motion. Differences in average articulatory dimensions between males and females can affect the overall size of the acoustic vowel space, potentially leading to a larger female acoustic vowel space.[14]
- **Interaction of Pitch and Articulation:** During speech, airflow from the lungs generates sound at the vocal folds or above the glottis, which is then modified by the pharynx, tongue, velum, and lips. Vowel quality is determined by tongue position and lip configuration, which influence the harmonics in the sound signal. For example, a high front tongue position with spread lips produces [i], while a high back tongue position with rounded lips produces [u].[14]
- **Voice Quality:** Beyond the pitch, male and female voices often exhibit differences in quality. Female voices may be perceived as smoother or more melodic, while male voices may sound rougher or more resonant. These differences can result from variations in vocal fold thickness, tension, and vocal tract resonance.[14]
- **Intonation Patterns:** Research shows that males and females may use different intonation patterns, particularly in terms of rising and falling pitch contours. Females may use more varied intonation patterns, with

greater pitch variation within sentences, while males may use flatter or more monotone intonation.[14]

- **Speech Rate:** On average, females tend to speak at a slightly faster rate than males. This difference in speech rate can influence overall communication dynamics, with faster speech potentially conveying enthusiasm or urgency, while slower speech may indicate thoughtfulness or deliberation.[14]
- **Use of Fillers and Discourse Markers:** There may be differences in the use of fillers (such as "um" and "uh") and discourse markers (such as "like" and "you know") between male and female speech. Research shows that females may use more fillers and discourse markers, possibly reflecting differences in conversational style or socialization.[14]
- **Pronunciation and Accent:** While pronunciation and accent can vary widely based on individual factors such as regional background and social identity, there may be subtle differences in the way males and females pronounce certain sounds or words. These differences can be influenced by factors such as social norms, education, and exposure to different speech communities.[14]

Chapter 3

Digital Signal Processing (DSP)

Digital Signal Processing has a huge impact on our daily lives. There are an endless list of devices that are influenced by the theory of Digital Signal Processing. In the modern era, we cannot describe our lives without this technology. Through the following subchapters, we will try to present information about Digital Signal Processing and principles.

3.1 Introduction of Digital Signal Processing(DSP)

We live in the age of technology where Digital Signal Processing and its technology have a huge role in this technology age and have a huge impact on our daily modern life. Because most of the devices that we are using in this technology era are built on Digital Signal Processing theory. Digital Signal Processors are successful implementations of this theory.

Currently, these devices are a huge part of our lives, a number of these devices are endless, and new devices are constantly expanding. Talking about Digital Signal Processing devices, without these devices, our daily life wouldn't be the same, we would lose access to digital/ Internet audio and digital/Internet video, as well as tools like CDs, DVDs, MP3 players, digital cameras, digital telephones, digital satellite and TV, and both wired and wireless network.

Thinking about losing voice recognition systems, speech synthesis systems, and image/video editing systems in today's era would make everything more challenging for humans. It's clear that without Digital Signal Processing, it would be harder for both humans and scientists to perform in different areas.[15] [5]

The concept of Digital Signal Processing is illustrated in the following picture.

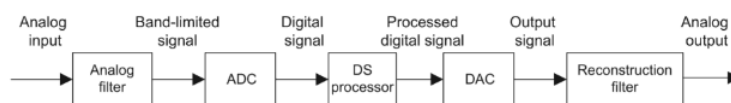


Figure 3.1: A digital processing scheme.[5]

In the diagram analog filter, an analog-to-digital conversion(ADC) unit, a digital signal(DS) processor, a digital-to-analog conversion (DAC) unit, and a reconstruction (anti-image) filter are illustrated.

3.2 Digital Signal Processing Basic Theory

The symbol of digital signal processing (DSP) technology lies in the digital signal processor (DSP). DSP is a field that utilizes computers or specialized digital processing equipment to process signals using numerical methods. It encompasses tasks such as data acquisition, signal transformation, analysis, synthesis, filtering, evaluation, and identification, all aimed at extracting information from signals.

Compared to traditional analog processing methods, digital processing offers unmatched advantages. Digital signal processing systems can handle both digital and analog signals. However, analog signals must first be converted into digital signals before they can be processed by a digital signal processing system.

The concept of typical digital signal processing flow is illustrated in the following picture.

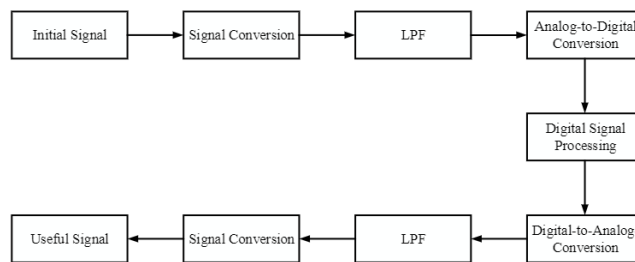


Figure 3.2: Digital Signal Processing Flow.[6]

The theory of digital signal processing involves several key aspects:

- **Pre-processing of analog signals:** This includes filtering out unwanted frequency components and noise in input analog signals to prevent spectral aliasing distortion after sampling.
- **Time domain sampling and recovery of analog signals:** This covers analog-to-digital conversion technology, sampling theorem, and quantization error analysis.
- **Analysis of time-domain discrete signals and systems:** This covers the representation and manipulation of signals, various transformations, and description and analysis of time-domain and frequency domains of time-domain discrete signals and systems.
- **Fast algorithms in digital signal processing:** This includes techniques such as fast Fourier transform and fast convolution.

- **Design and implementation** of analog filters and digital filters.
- **Multi-sampling-rate signal processing technology:** This involves the basic principle of sampling rate conversion systems and their efficient implementation methods.

A quasi-signal processing system can only perform conventional simple processing on signals, while digital signal processing utilizes numerical operations to implement signal processing, allowing for more complex operations to be performed using computers. Therefore, digital signal processing applications have a wider range of possibilities.[6]

3.3 Digital Signal Processing systems

A Digital Signal Processing(DSP) system encompasses devices or setups that perform digital signal processing operations. This can involve software, such as algorithms running on a computer, or hardware, such as circuits or specialized chips. In many cases, it's a combination of both.

Digital Signal Processing(DSP) systems find applications across various fields:

- **Audio and speech processing:** Used to enhance sound quality, perform speech recognition, and create digital synthesizers.
- **Image and video processing:** Includes tasks such as image enhancement, restoration, recognition, and digital video broadcasting.
- **Radar and sonar:** Utilized for remote sensing and extracting useful information from signals in radar and sonar systems.
- **Telecommunications:** A Digital Signal Processing(DSP) is employed for data compression and decompression, error detection and correction, and modulation and demodulation in telecommunications systems.
- **Biomedical engineering:** Used in medical image processing, as well as for signal processing in electrocardiograms (ECG) and electroencephalograms (EEG).
- **Seismology:** A Digital Signal Processing(DSP) is utilized in devices for processing data from seismic instruments to interpret the status of the Earth's interior.

Apart from these, Digital Signal Processing(DSP) plays a significant role in improving the quality of audio recordings, creating new sounds, and correcting problems with audio signals.

Digital Signal Processing(DSP) is used in audio applications:

Chapter 4

Automatic Speech Recognition

Humans always had the desire to automate even small tasks, since a century ago they worked ways for this but it wasn't until recent years. Speech is the main pillar of the way people communicate with each other. Speech technology is transforming our lives and becoming one of the primary means. Through our daily life for daily use which humans interact with devices. Automatic speech recognition (ASR) is an important technology to enable and improves the human-human and human-machine interactions. Automatic speech recognition (ASR) is one of the important bridges between this human-machine interaction. This has been studied for five decades because it is always seen as an important bridge for communication. However, even though it was seen as important, it never became significant due to limitations. This was mostly because the technology at that time was quite insufficient for real usage conditions. But in recent years, these problems progressed after improvements have been made in several key areas. In this chapter, we introduce the main application areas of ASR systems, describe their basic architecture, and then error rate analysis of current ASR systems.[16]

4.1 Introduction of ASR

Automatic Speech Recognition, also known as Speech Recognition, is a technology that processes and turns human speech into text. Thanks to this advanced technology, it is not only just an application for speech-to-text, but it also provides usage from voice-enabled assistants and transcription services to accessibility features for those with disabilities. With help from these systems, daily life became easier not only in the different work areas but also gives help to disabled people who help their daily lives to go smoothly.

If we look deeper into the core of ASR, we will understand that it involves the conversion of audio signals into linguistic units using algorithms and models that capture the complexity of language, accents, and various speech patterns. Today, modern ASR systems take advantage of machine learning. These systems are particularly keen on deep learning techniques, to improve accuracy and adapt to the spoken language.

Over the years, ASR technology has grown enormously, with the help of deep learning techniques. Now it is not only simply machines that can

representation is the starting point for most applications. From this point, other representations are obtained by digital processing.[7]

The Speech Chain is illustrated in the following picture.

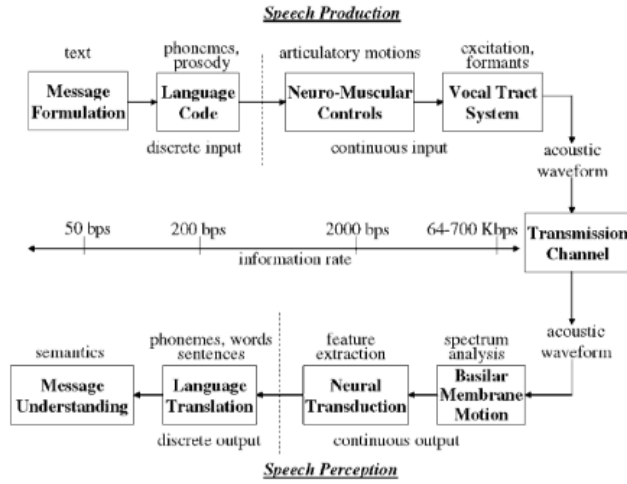


Figure 4.1: The Speech Chain: from message to the speech signal to understanding.[7]

4.1.2 Speech Coding

Most widespread applications of digital speech processing technology are perhaps seen in the areas of digital transmission and storage of speech signals. Since the goal is to compress the digital waveform representation of speech into a lower bit-rate representation, the centrality of the digital representation is obvious. The process is referred to as “speech coding” or “speech compression” commonly.[7]

The speech coding block diagram is illustrated in the following picture.

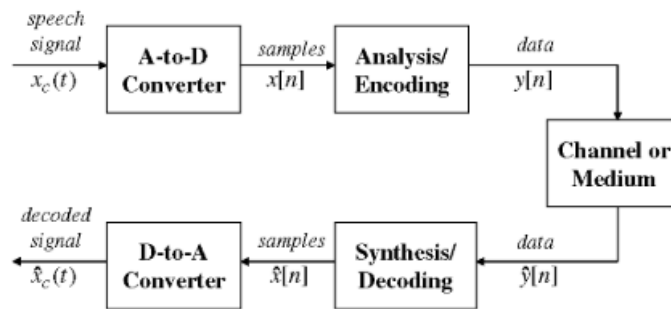


Figure 4.2: Speech coding block diagram — encoder and decoder.[7]

The illustrated figure shows a block diagram of a generic speech encoding/decoding (or compression) system. The lower path in the figure shows the decoder associated with the speech coder. The D-to-A decoder is often called

transactions over the internet, handling call center help desks and customer care applications, serving as the voice for providing information from handheld devices such as foreign language phrasebooks, dictionaries, crossword puzzle helpers, and as the voice of announcement machines that provide information such as stock quotes, airline schedules, updates on arrivals and departures of flights, etc. Another important application is in reading machines for the blind or hearing machines for people who have hearing problem.[7]

4.2 Feature Extraction

The simplicity of human speech belies the complexity of the task, which could explain why speech is highly sensitive to diseases associated with the nervous system.

Several successful attempts have been made to develop systems capable of analyzing, classifying, and recognizing speech signals. Both hardware and software designed for these tasks have found applications in various fields, including healthcare, government sectors, and agriculture. Speaker recognition refers to the ability of software or hardware to receive speech signals, which can identify the speaker present in the speech signal, and then recognize the speaker. Speaker recognition mimics the process undertaken by the human brain. It begins with speech as input to the speaker recognition system. Overall, the speaker recognition process consists of three main steps: acoustic processing, feature extraction, and classification/recognition.

Before extracting the essential attributes in the speech and identification, the speech signal must undergo noise removal. Feature extraction aims to represent a speech signal with a predetermined number of components, as dealing with all the information in the acoustic signal is impractical, and some of it is irrelevant to the identification task.

Feature extraction involves transforming the speech waveform into a parametric representation at a lower data rate for subsequent processing and analysis, commonly known as front-end signal processing. This process aims to convert the processed speech signal into a concise yet meaningful representation that is more discriminative and reliable than the original signal. The quality of subsequent features, such as pattern matching and speaker modeling, is notably influenced by the quality of the front-end.

Thus, the quality of the features directly impacts the accuracy of classification. In current automatic speaker recognition (ASR) systems, the focus of feature extraction is to find a representation that remains reliable across various conditions of the same speech signal, even with changes in environmental conditions or speakers, while preserving the informative aspects of the speech signal. Feature extraction methods typically yield a multidimensional feature vector for each speech signal. Various parametric representations are available for this process, including perceptual linear prediction (PLP), linear prediction coding (LPC), and mel-frequency cepstrum coefficients (MFCC), with MFCC being the most widely used.

Feature extraction is crucial in speaker recognition as speech features

compared to low-frequency formants, emphasis is placed on high frequencies to achieve similar amplitudes for all formants.

After windowing, the Fast Fourier Transform (FFT) is applied to each frame to obtain its power spectrum. Subsequently, filter bank processing is carried out on the power spectrum using a mel-scale.

The Discrete Cosine Transform (DCT) is then applied to the log-domain power spectrum to calculate the MFCC coefficients.

The formula used to calculate the mel frequency for any given frequency f is:

$$mel(f) = 2595 \cdot \log_{10} \left(1 + \frac{f}{700} \right) \quad (4.1)$$

where $mel(f)$ is the frequency (mels) and f is the frequency (Hz). The MFCCs are calculated using this equation :

$$C_n = \sum_{k=1}^N \log(S_k) \cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{k} \right] \quad (4.2)$$

where k is the number of mel cepstrum coefficients, $[S_k]$ is the output of filterbank, and $[C_n]$ is the final mfcc coefficients.

The block diagram of the MFCC processor is illustrated in the following picture.

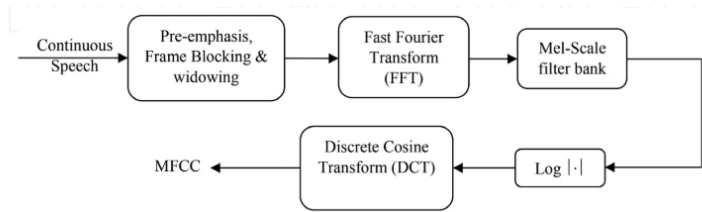


Figure 4.4: Block diagram of MFCC processor.[8]

This section outlines the procedures and steps involved in obtaining the necessary coefficients. MFCC is particularly effective in representing the low-frequency region compared to the high-frequency region. Consequently, it can accurately compute formants within the low-frequency range, describing the resonances of the vocal tract. It's widely acknowledged as a front-end process for typical Speaker Identification applications due to its reduced susceptibility to noise interference, minimal session-to-session variation, and ease of implementation. Moreover, it serves as an excellent representation of stable and consistent source characteristics such as music and speech. Additionally, it can capture information from sampled signals with frequencies up to 5 kHz, encompassing the majority of energy in sounds generated by humans.

Cepstral coefficients have been found to be accurate in certain pattern recognition problems related to human voice, and they are extensively used in

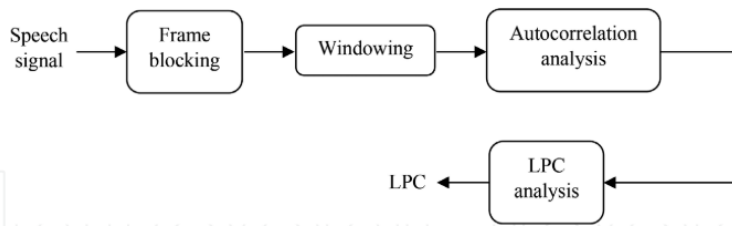


Figure 4.5: Block diagram of LPC processor.[8]

After windowing the signal, each frame is autocorrelated, with the highest autocorrelation value determining the order of the linear prediction analysis. Then, LPC analysis is performed, converting each frame of autocorrelations into a set of LPC parameters, which consists of the LPC coefficients.

Linear predictive analysis efficiently extracts vocal tract information from speech signals. It is renowned for its computational speed and accuracy, providing excellent representation for stable and consistent source behaviors. Additionally, LPC is commonly utilized in speaker recognition systems to extract vocal tract properties, offering very accurate estimates of speech parameters and demonstrating comparative computational efficiency.

However, traditional linear prediction techniques may suffer from aliased autocorrelation coefficients and high sensitivity to quantization noise, making them less suitable for generalization[8].

■ 4.2.3 Linear prediction cepstral coefficients (LPCC)

Linear prediction cepstral coefficients (LPCC) are derived from the spectral envelope calculated by linear predictive coding (LPC). They represent the Fourier transform of the logarithmic magnitude spectrum of LPC. Cepstral analysis is widely used in speech processing due to its ability to accurately represent speech waveforms and characteristics using a compact set of features.

Rosenberg and Sambur observed that adjacent predictor coefficients in LPC are highly correlated. Therefore, representations with less correlated features, such as LPCC, are more efficient. The relationship between LPC and LPCC was initially derived by Atal in 1974. In theory, it is relatively straightforward to convert LPC to LPCC, particularly in the case of minimum phase signals.[8]

■ Algorithm description, strength, and weaknesses

In speech processing, LPCC, like LPC, is calculated from sample points of a speech waveform. The horizontal axis represents time, while the vertical axis represents amplitude.

The Block diagram of LPCC processor is illustrated in the following picture.

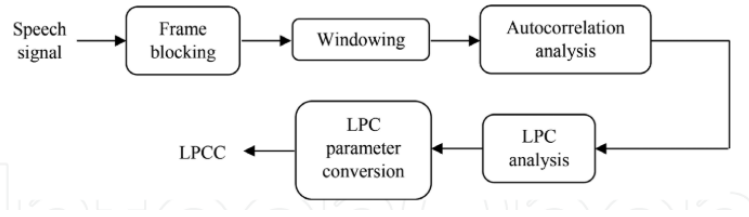


Figure 4.6: Block diagram of LPCC processor.[8]

LPCC can be calculated using:

$$C_m = a_m + \sum_{k=1}^{m-1} \left[\frac{k}{m} \right] c_k a_{m-k} \quad (4.5)$$

where $[a_m]$ is the linear prediction coefficient, $[C_m]$ is the cepstral coefficient.

LPCC exhibits lower susceptibility to noise and generally yields lower error rates compared to LPC features. However, as the order of cepstral coefficients increases, there is a significant increase in variance. LPCC estimates are also known for their high sensitivity to quantization noise. Additionally, cepstral analysis of high-pitched speech signals often results in limited source-filter separability in the quefrequency domain. Lower-order cepstral coefficients are sensitive to spectral slope, whereas higher-order coefficients are more sensitive to noise.[8]

■ 4.2.4 Line spectral frequencies (LSF)

Each line of the Line Spectral Pairs (LSP) corresponds to a line spectral frequency (LSF), which characterizes the resonance patterns in the interconnected tube model of the human vocal tract. This model incorporates the nasal cavity and mouth shape, forming the basis for the physiological significance of linear prediction representation. The two resonance situations defined by LSF correspond to the vocal tract either being fully open or fully closed at the glottis. These situations result in two groups of resonant frequencies, determined by the number of interconnected tubes. The odd and even line spectra represent the resonances of each situation, woven into a singularly rising group of LSF.

The LSF representation, proposed by Itakura, serves as an alternative to linear prediction parametric representation. In the field of speech coding, it has been found that LSF representation offers improved quantization features compared to other linear prediction parametric representations (such as LAR and RC). LSF representation can reduce bit-rate by 25–30% for transmitting linear prediction information without compromising the quality of synthesized speech. Besides quantization, the LSF representation of the predictor is also suitable for interpolation. The diagonal sensitivity matrix linking LSF-domain squared quantization error to the perceptually relevant log spectrum theoretically inspires this capability.[8]

■ Algorithm description, strength, and weaknesses

The block diagram of the LSF processor is illustrated in the following picture.

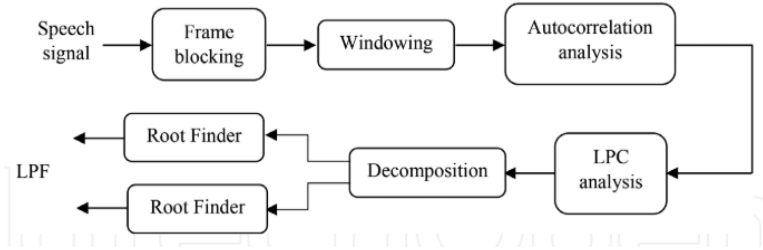


Figure 4.7: Block diagram of LSF processor.[8]

LSF finds its primary application in speech compression, although its utility extends to speaker recognition and speech recognition. Additionally, LSF has been explored in other domains, such as musical instrument recognition and coding, animal noise identification, recognizing individual instruments, and financial market analysis. The key advantages of LSF lie in their ability to localize spectral sensitivities, characterize bandwidths and resonance locations, and emphasize spectral peak locations. In many cases, the LSF representation offers a nearly minimal dataset for subsequent classification tasks.

Due to its ability to represent spectral shape information at a lower data rate compared to raw input samples, LSF can lead to complexity reduction when processing and analyzing methods are carefully applied in the LSP domain, compared to techniques operating on raw input data. LSF plays a crucial role in transmitting vocal tract information from speech coder to decoder, owing to its excellent quantization properties. The generation of LSP parameters can be achieved through various methods, ranging from standard root-solving techniques to more sophisticated methods, often performed in the cosine domain.[8]

■ 4.2.5 Discrete wavelet transform (dwt)

The theory of Wavelet Transform (WT) revolves around signal analysis using varying scales in both the time and frequency domains. Introduced by Jean Morlet with the support of theoretical physicist Alex Grossmann, wavelet transform enables the identification of high-frequency events with enhanced temporal resolution. A wavelet is a waveform of limited duration that averages to zero, and many wavelets exhibit orthogonality, making them ideal for compact signal representation.

WT is a powerful signal processing technique capable of efficiently representing real-life non-stationary signals. It can extract information from transient signals simultaneously in both time and frequency domains.

The Continuous Wavelet Transform (CWT) decomposes a continuous-time function into wavelets. However, it suffers from information redundancy, and computing all possible scales and translations of CWT requires significant computational effort, limiting its practical use.

relevant information from speech. Based on the nonlinear bark scale and was originally designed for use in speech recognition tasks to remove speaker-dependent features. PLP provides a representation that resembles a smoothed short-term spectrum, equalized and compressed to mimic human hearing, which is making it to similar MFCC.

In the PLP technique, several key features of human hearing are emulated, and the resulting auditory-like spectrum of speech is approximated by an autoregressive all-pole model. PLP achieves reduced resolution at high frequencies, typical of auditory filter bank-based approaches while producing orthogonal outputs similar to cepstral analysis. It employs linear predictions for spectral smoothing, hence the name "perceptual linear prediction". PLP combines both spectral analysis and linear prediction analysis.

■ Algorithm description, strength, and weaknesses

To compute the PLP features, the speech signal undergoes several steps. First, it is windowed using a Hamming window, and then the Fast Fourier Transform (FFT) is applied to compute the square of the magnitude, yielding power spectral estimates.

Next, a trapezoidal filter is used at 1-bark intervals to integrate the overlapping critical band filter responses in the power spectrum. This compression effectively narrows the higher frequencies into a band.

Following this, symmetric frequency domain convolution on the bark-warped frequency scale allows low frequencies to mask the high frequencies, smoothing the spectrum simultaneously. The spectrum is then pre-emphasized to mimic the uneven sensitivity of human hearing across different frequencies.

Subsequently, spectral amplitude compression reduces the amplitude variation of the spectral resonances. An Inverse Discrete Fourier Transform (IDCT) is performed to obtain the autocorrelation coefficients.

Finally, spectral smoothing is carried out by solving the autoregressive equations, and the autoregressive coefficients are converted to cepstral variables.[8]

The Block diagram of PLP processor is illustrated in the following picture.

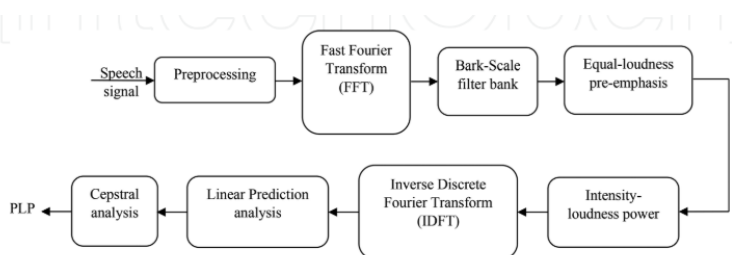


Figure 4.9: Block diagram of PLP processor.[8]

Figure illustrates the PLP processor, detailing the steps involved in obtaining the PLP coefficients. PLP demonstrates low sensitivity to spectral tilt, which aligns with studies suggesting its relative insensitivity to phonetic judgments of spectral tilt. However, PLP analysis depends on the overall spectral

Chapter 5

Software tools

This chapter is dedicated to software tools that are used in this research. These engines are chosen for their unique strengths and play a pivotal role in integrating complex algorithms and functionalities. Through the following subchapters, we will try to present information about the tools and their distinct roles and capabilities.

5.1 Background noise-cleaning tools

Background cleaning tools play a vital role in ensuring the reliability and effectiveness of ASR systems in real-world settings. These tools are crucial for ASR systems because they improve accuracy by reducing background noise and enhancing the Signal-to-Noise Ratio (SNR). These tools enable ASR systems to perform well in noisy environments, ensuring accurate transcription and a better user experience. Additionally, they help mitigate errors caused by background noise, resulting in clearer speech input and improved recognition performance.

Now, let's check the background noise-cleaning tools used in this research.

5.1.1 Descript

Descript offers tools to record, edit, transcribe, collaborate, and share videos and podcasts, it gives an entirely new approach to editing audio. Descript works simply, it's a vastly different way of doing things compared to traditionally complex audio tools. First, with artificial intelligence, it automatically transcribes all your uploaded audio, then you edit your recordings by simply highlighting and deleting or moving any words or passages in a text editor.

Initially, Descript focused on audio content. The platform's core capability was providing a document-style interface for professionally enhancing and editing audio recordings by interacting with auto-generated transcripts.[17] These are the main features of Descript used in this research:

- **Overdub (text-to-speech):** Descript can create a voice model of the user and add synthetic voice to content (text-to-speech) by editing the transcript. It has professional voice blending and multiple voices for different contexts.

- **Studio quality sound:** Descript supports professional audio quality without requiring expensive hardware. It has noise removal, speech enhancement, acoustic echo cancellation, and sound effects.

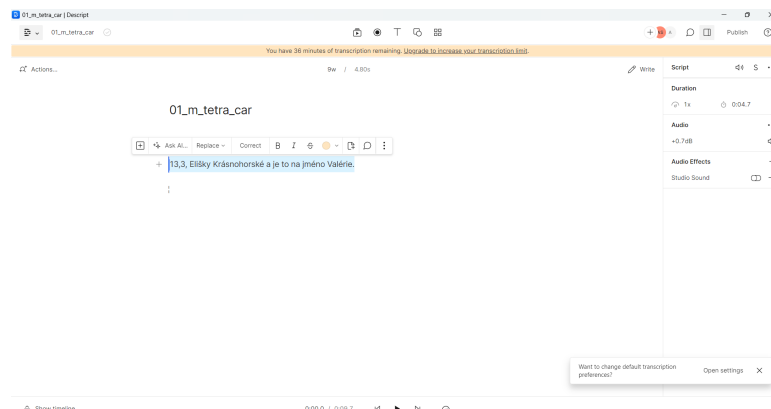


Figure 5.1: Descript.

We can say that, Descript makes it much faster and easier to use professional audio tools. It hides complexity from the user, but it also lets you go deep when needed.[17]

5.1.2 iZotope RX 10

iZotope RX is an award-winning audio restoration plug-in suite designed to repair noisy or damaged audio, remove background noise, and more. Specifically, iZotope RX 10 is the industry's most intuitive and intelligent audio repair suite for restoring, cleaning up, and improving recordings in post-production, music, and content creation.

From analyzing audio capture and production issues to providing processing solutions, RX 10 empowers to achieve upgrades from poor audio files that were once unfixable and deliver reliably clear sound quality.[18]

These are the main features of iZotope RX 10 that were used in this research:

- **Repair Assistant:** Repair Assistant Plug-ins are time-saving tools in editing. The Repair Assistant plug-in uses machine learning to find and fix audio issues quickly without leaving your DAW. The Assistant automatically recognizes specific problems and intelligently proposes a repair chain that you can modify to your liking with easy-to-use dials. This plug-in was built from the ground up.[18]
- **De-hum:** With Dynamic Adaptive Mode in De-hum, you can save time removing unwanted hum. You can get rid of hums and buzzes on the fly, without having to spend time learning the noise profile of your audio. Eliminating electromagnetic interference or other complex noise that changes pitch can be done automatically without sacrificing quality.[18]

- **De-noise:** Denoisers are reducing and eliminating steady-state background noise. They can be based on FFT with thousands of bands, or a simple crossover with just a few bands, and are sometimes designed for a specific use case, such as vocals.[18]
- **De-click:** Declickers are minimizing and eliminating disruptive clicks and pops in audio recordings. These can be caused by anything from dust and scratches on an old record, a CD skipping on playback, or even mouth clicks and lip smacks from a voiceover.[18]
- **Spectral Recovery:** Spectral Recovery is a tool for bringing life back to thin-sounding audio to match the rest of your productions. It improves upon the quality of re-synthesized upper frequencies and can add missing lower frequencies, too.[18]

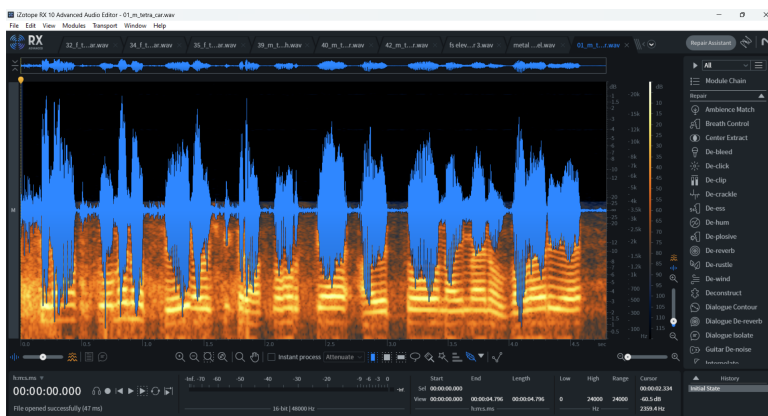


Figure 5.2: iZotope RX 10



Figure 5.3: iZotope RX 10



Figure 5.4: iZotope RX 10

5.1.3 Audacity

Audacity is the world’s most popular free software for recording and editing audio. Audacity is a free and open-source digital audio editor and recording application software, available for Windows, macOS, Linux, and other Unix-like operating systems.

Now we will see how to reduce noise on audacity. Noise reduction allows you to remove or reduce background noise from your recordings. First, we need to open an audio file on Audacity. We will be using our example file for demonstration. We will start by selecting a section of the audio containing only the noise that we want to reduce or remove. Try the beginning or the end of your recording or look for a good pause somewhere in track. To get a good analysis of the noise, we should select enough of the audio. Now we go into the effect menu and scroll down to noise reduction. Then click on get noise profile. This will close the dialog box and create a sample of noise. Audacity uses that sample to analyze and extract the noise from your signal.[19]

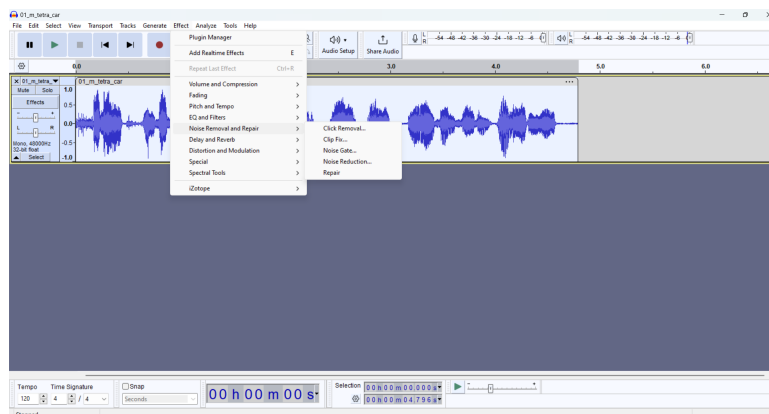


Figure 5.5: Audacity

Now we need to select your entire timeline by double-clicking on the audio

track. Then go back into the effect menu and select noise reduction again.

There are **three adjustable settings** that you can use to fine-tune noise reduction. Each of these controls affects how the noise is reduced. Raising them too high will remove more noise but at the expense of some of the audio quality. Setting them too low will keep more of the desired audio signal but it also retains some of the noise. So we adjust these to the level that is satisfactory to us.

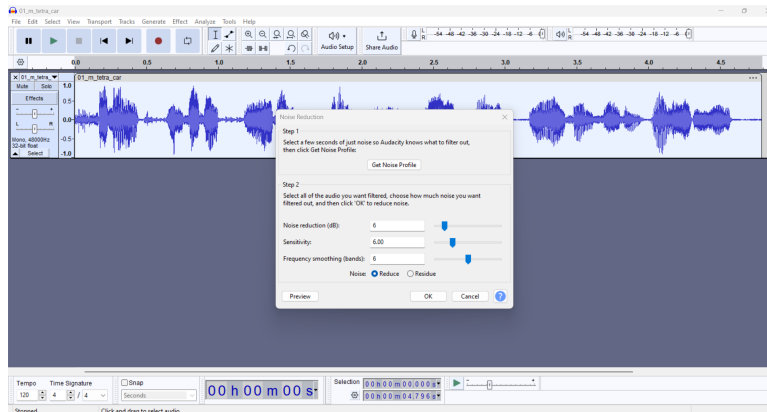


Figure 5.6: Audacity

- The first control is noise reduction amount. This setting reduces the identified noise in decibels. We adjust it to a point where we feel enough noise has been removed. We can click on the preview button to hear the results of our settings as we alter them.
- The sensitivity setting controls how much of the audio is considered noise. Setting the sensitivity too low can result in higher frequency distortions in the audio. Therefore set this to the lowest level that achieves a satisfactory amount of noise removal without introducing distortions.
- The final control is frequency smoothing. This control helps smooth out possible distortions that occur due to increasing the frequency range of the noise being reduced. A higher setting will make your audio signal less clear so keep it low or even off.

The default settings work well for basic noise removal. Another way to hear the effect of your setting is to listen to only the parts of the signal being removed. We can do this by selecting the residue option in the noise setting. Select this option and press preview. This is useful for finding the optimum settings that do not damage the audio. If you can hear recognizable bits of desired sound in the residue, you have likely set noise reduction or sensitivity too high.[19]

5.1.4 Audacity x RX 10

After we purchased and installed iZotope RX 10, powerful audio repair tools became available as plugins in Audacity. If we open Audacity, we will notice

that there are plugins that we can use and we can find these plugins listed in the Effects tab. There's an option here to see all the plugins they have. You can use and then there's an option to even add or remove plugins according to your preference. For now, we decided to use only RX 10 plug-ins.[19]

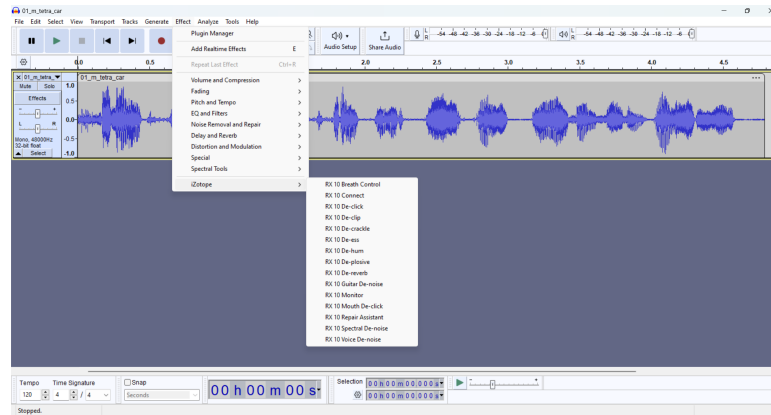


Figure 5.7: Audacity x RX 10

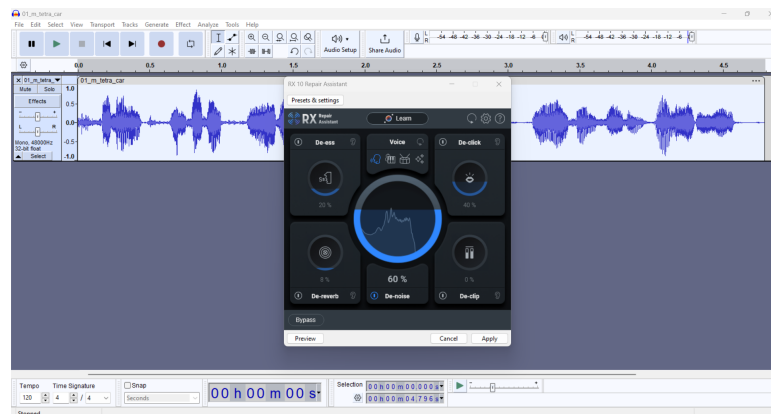


Figure 5.8: Audacity x RX 10

5.2 Speech-to-Text tools

Speech-to-text tools convert spoken language into written text, offering convenience and efficiency in various applications. These tools utilize advanced algorithms and machine-learning techniques to accurately transcribe audio content. They support multiple languages and dialects.

Now, let's check the Speech-to-Text tools used in this research.

5.2.1 Whisper

Whisper is a general-purpose automatic speech recognition (ASR) system. It is trained on a large dataset of diverse audio and is also a multitasking model that can perform multilingual speech recognition, speech translation, and

language identification. A system trained on 680,000 hours of data collected from the web. The use of such a large and diverse dataset leads to improved robustness to accents, background noise, and technical language. Moreover, it enables transcription in multiple languages, as well as translation from those languages into English.[20]

The Whisper architecture is a simple end-to-end approach, implemented as an encoder-decoder Transformer. Input audio is split into 30-second chunks, converted into a log-Mel spectrogram, and then passed into an encoder. A decoder is trained to predict the corresponding text caption, intermixed with special tokens that direct the single model to perform tasks such as language identification, phrase-level timestamps, multilingual speech transcription, and to-English speech translation. Whisper's audio dataset is non-English, and it is alternately given the task of transcribing in the original language or translating to English. This approach is particularly effective at learning speech-to-text translation and outperforms the supervised SOTA on CoVoST2 to English translation zero-shot.[20]

The installation process of Whisper AI is a quite seamless process, deployed seamlessly across both local and cloud platforms. The installation process is streamlined, facilitating scalability and adaptability to the evolving demands of data analysis endeavors.

■ Local platform

To get Whisper AI working on your computer, we need to install **five** different items. First, we need to install Python. Python is the programming language that Whisper AI uses. There are a few different versions. Whisper AI works from version 3.7 all the way up to 3.10. It currently does not work on 3.11. You can choose your operating system. I'm running a Windows machine, so I have the Windows version.

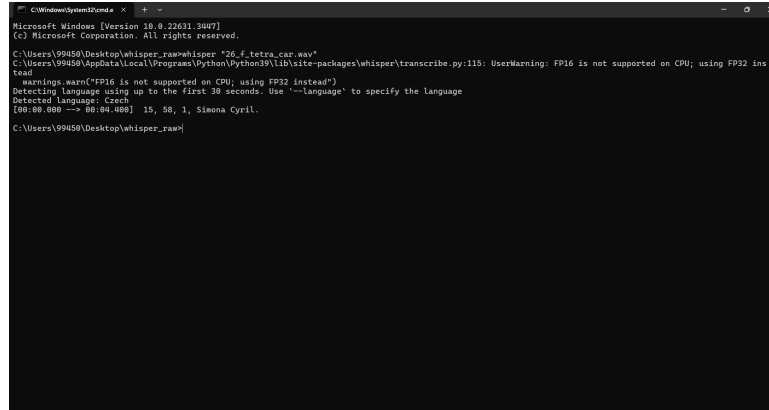
Next, we need to install PyTorch. PyTorch is a machine-learning library. We need to install the current stable version. You need to choose your operating systems, then package type, and select PIP since we are using Python, for the language, we'll use Python. And we can choose the computing platform. Since we don't have a high-powered GPU, we selected CPU but this doesn't go as quickly as a dedicated graphics card. On command prompt, we need to press right mouse button, and that will paste the command that we copied for installing PyTorch.

Now, we need to download a package manager called Chocolatey. On PowerShell, we need to select run as administrator. This now opens up PowerShell and we need to press right mouse button, and that will paste the command that we copied for installing Chocolatey.

Now we need to use the Chocolatey package manager to install FFMPEG, and we're going to use FFMPEG to read the different audio files, so whether it's a WAV file or whether it's an MP3. On PowerShell, we can install the package manager.

Finally, now in command prompt in administrator mode, install the final

item, Whisper AI. To install it, type in `pip install`, and next, type in `OpenAI-Whisper`, This will install Whisper AI.



```

Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\99450\Desktop\whisper_rax\whisper > %0 f letis_csr.wav
C:\Users\99450\AppData\Local\Programs\Python\Python39\lib\site-packages\whisper\transcribe.py:115: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
Detecting language using up to the first 30 seconds. Use '--language' to specify the language
Detected language: Czech
[00:00:00 -> 00:04:00] 15, 58, 1, Simona Cyril.

C:\Users\99450\Desktop\whisper_rax>

```

Figure 5.9: Whisper AI(local)

■ Cloud platform

You can install Whisper directly on your computer. But you do need a somewhat capable computer. So instead, we can use something called Google Colaboratory. This allows you to run code directly in your web browser. So it doesn't matter what type of PC you have. To use Google Colaboratory, head to Google Drive. You'll need a Google account. On Google Drive, in the top left-hand corner, let's click on the New button. And at the very bottom, let's click on More, and then go down to Connect More Apps. At the top of this dialog, let's click into the search field, and here, type in Google Colaboratory and then search. We see the result for Colaboratory and there, we install it. With this Google Colaboratory was connected to Google Drive.

After this, now go back to the top left-hand corner. If we click on the New button again. Then go down to More. Here, you should now see an option for Google Colaboratory. If we click on this one, this drops us into the Google Colaboratory space. First, in the top left-hand corner, we can give our file a name. Next, let's click on the menu titled Runtime, and right here, there's the option for Change Runtime Type. Let's click on that, and that opens up this dialog where we can choose the hardware accelerator. We selected GPU(or graphics card), and it turns out that that graphics cards run these models extremely well. After saving this, we need to install Whisper AI. You simply need to copy and paste code from the GitHub source. Then we will install something called ffmpeg. This allows us to work with audio and video files. This is not installing anything on your computer, this is installing it all to the Google Colaboratory. After finishing the installation, over the left-hand side, let's click on this Folder icon. And you can now drag in an audio file or a video file that you would like to transcribe.

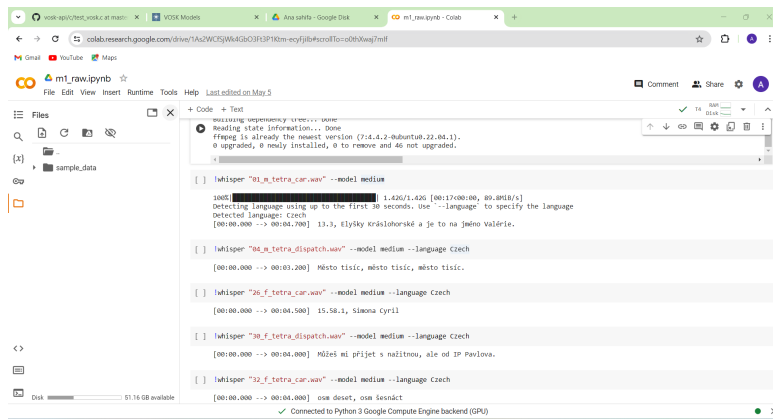


Figure 5.10: Whisper AI(cloud)

5.2.2 Vosk

Vosk is an offline open-source speech recognition toolkit. It enables speech recognition for 20+ languages and dialects (English, Indian English, German, French, Spanish, Portuguese, Chinese, Russian, Turkish, Vietnamese, Italian, Dutch, Catalan, Arabic, Greek, Farsi, Filipino, Ukrainian, Kazakh, Swedish, Japanese, Esperanto, Hindi, Czech, Polish).

Vosk models are small (50 Mb) but provide continuous large vocabulary transcription, zero-latency response with streaming API, reconfigurable vocabulary, and speaker identification.[15]

To install the environment, you'll need a couple of dependencies. The main one is pyaudio, which is installed first. Then you need to install the model that you want to use. We installed English and Czech models. For all these models, we contained them in the same folder. Next, we create a new file for this. We have installed test examples for real-time transcribing and audio files. Note that for audio files, it accepts WAV files.[21]

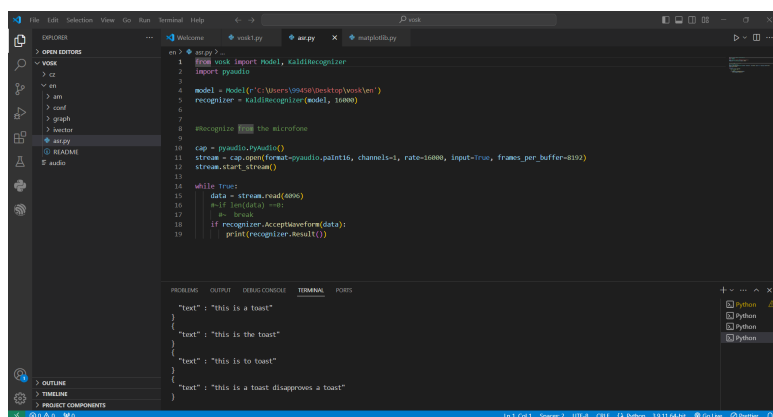


Figure 5.11: Vosk

Chapter 6

Simulated experiments

This chapter is dedicated to the simulation of experiments in this research. Through the following subchapters, we will try to present our approach, our data, and our test setup. We will talk about considering the ethical and privacy considerations for our experiments.

6.1 Methodolgy

Our audio analysis strategy employs two critical functions: background noise cleaning and Speech-to-Text conversion. With the help of an intuitive interface and advanced algorithms, we will not only adeptly eliminate unwanted background noise but also facilitate precise and efficient conversion of spoken words into text. We will compare our raw data with trained data. The reason behind this analysis is, we want to see how different tools can work together. It makes sense to push the idea that Data cleaning and preprocessing are imperative to guarantee the dataset's quality and reliability. But how this improved quality help Speech-to-Text tools? Does it make it better or worse? One of our tools for background noise removal; Descript works as both: background noise cleaning and Speech-to-Text conversion. It gives us inspiration that, both of these tools can work together. During our analysis, we will try this dual functionality to clarity and accuracy of our audio data, and analyze outcomes.

Our approach's flow is illustrated in the following picture.

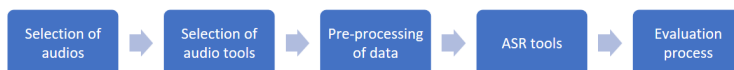


Figure 6.1: Our approach

Our approach to identifying the optimal Automatic Speech Recognition (ASR) application tailored is a five-step methodology devised for this research. First, we selected audio files and data for this experiment. After choosing this data, we looked for tools for repairing or cleaning these audio data sets. After we pre-processed these data, we used these audio files on Speech-to-

5 unique and different speaking styles, across different environments. The sentences on these audio files contain short and long words, sentences with numbers, or just numbers. The audio samples were carefully chosen to test the ASR system's performance.

For female speech:

Female audio	Sentences	Wordcount
Female voice 1	15, 58, 1, Simona, Cyril.	3 numbers and 2 words
Female 2	Můžeš mi přijet na Žitnou, ale od IP Pavlova?	9 words
Female 3	Osm deset, osm šestnáct	4 words
Female 4	Takže čas ověření 8.48	2 numbers and 3 words
Female 5	6, 7, 5, 7, 7, 4	6 numbers

Table 6.2: Female speech database

The table illustrates 5 different female voices we used, 5 female speakers speak 5 unique and different speaking styles, across different environments. The sentences on these audio files contain short and long words, sentences with numbers, or just numbers. The audio samples were carefully chosen to test the ASR system's performance.

As we know our experimental data, we can discuss data preprocessing now. Cleaning and preprocessing of data are essential steps to guarantee the dataset's quality and reliability. We took measures to address issues such as missing values, outliers, and inconsistencies, establishing a robust foundation for subsequent analysis. Additionally, standardization and transformation procedures were applied to enhance comparability and align the data with our research objectives.

6.3 Test setup

Our test setup follows up after data preprocessing. As the next steps, we are following up on ASR software tools. The basic principle is finding the best possible optimal match between these tools.

Using the simple table, we will match audio tools with ASR tools.

Here is the table of audio tools:

Audio tools
Descript
iZotope RX 10
Audacity
Audacity x RX 10

Table 6.3: Audio tools

These are the tools that are used for improving data quality. With this improved quality, these data are going to test ASR tools.

Next are ASR tools, with another table, we are going to check them one more time before matching:

ASR tools
Descript
Whisper AI (local)
Whisper AI (cloud)
Vosk

Table 6.4: ASR tools

Now, that we introduced our tools here is our test setup for this experiment. As can be seen, we have four audio tools and four ASR tools which means that for every ASR tool we will use audio tools one by one. After completing our evaluation, we will compare results and analyze outcomes.

In the next tables, we will illustrate our experiment setup:

First Set:

First Experiment Setup
Descript x Descript
Descript x iZotope RX 10
Descript x Audacity
Descript x Audacity x RX 10

Table 6.5: First Experiment Setup

Second Set:

Second Experiment Setup
Whisper AI (local) x Descript
Whisper AI (local) x iZotope RX 10
Whisper AI (local) x Audacity
Whisper AI (local) x Audacity x RX 10

Table 6.6: Second Experiment Setup

Third Set:

Third Experiment Setup
Whisper AI (cloud) x Descript
Whisper AI (cloud) x iZotope RX 10
Whisper AI (cloud) x Audacity
Whisper AI (cloud) x Audacity x RX 10

Table 6.7: Third Experiment Setup

Fourth Set:

Fourth Experiment Setup
Vosk x Descript
Vosk x iZotope RX 10
Vosk x Audacity
Vosk x Audacity x RX 10

Table 6.8: Fourth Experiment Setup

After these software modules undergo testing for speech recognition, we will conclude the test with the comparison of outcomes. For this evaluation, we will use the word error rate (WER). In speech recognition, the word error rate (WER) is a widely adopted parameter and evaluation standard. The speech recognition program must automatically perform substitutions, deletions, or insertions of specific words to ensure coherence between the recognized word sequence and the standard word sequence.

The Word Error Rate (WER) meticulously assesses these discrepancies by comparing the reference transcript with the expected output word by word. WER considers three types of errors:

- **Insertion:** Additional words in the output that are not present in the transcript.
- **Deletion:** Absence of words in the output that exist in the transcript.
- **Substitutions:** Incorrectly translated words in the output that replace terms in the transcript.

WER is calculated by dividing the total number of inserted, replaced, and deleted words by the percentage of the number of words in the standard word sequence, as outlined below.

The Word Error Rate (WER) is given by:

$$\text{WER} = \frac{S + I + D}{N} \times 100\%$$

In the context of the formula, where S represents the replacement word, I stands for the inserted word, D indicates the deleted word, and N is the total number of words, discrepancies can arise during the conversion of spoken words into text by voice recognition software.

Automatic Speech Recognition (ASR) with a lower WER generally demonstrates greater accuracy in speech identification and a higher WER often indicates lower accuracy in Automatic Speech Recognition (ASR).

6.4 Calculation of ASR results

The Word Error Rate (WER) stands as a key metric, between a reference transcription and the output of the system for our experiment. To simplify this evaluation process, we used the Jiwer library which is a powerful and

Chapter 7

Practical experiment

This chapter is dedicated to the experiments and their results in this research. Through the following subchapters, we presented how to implement ASR in different environments, and the results after experiments and discussion about key findings. Chapters end with suggestions about future research directions in ASR.

7.1 Guidelines for implementing ASR in environments

This section provides insights into the way we implemented Automatic Speech Recognition (ASR) on Descript, Whisper AI (local and cloud), and Vosk.

■ Speech-to-text on Descript

Our first tool is Descript. To begin this, we simply need to create an account on Descript, then open a new project and choose what type of project it is. In our case, we chose an audio project because of our data. After adding your audio file to your workspace, the languages for transcribing appear on the screen.

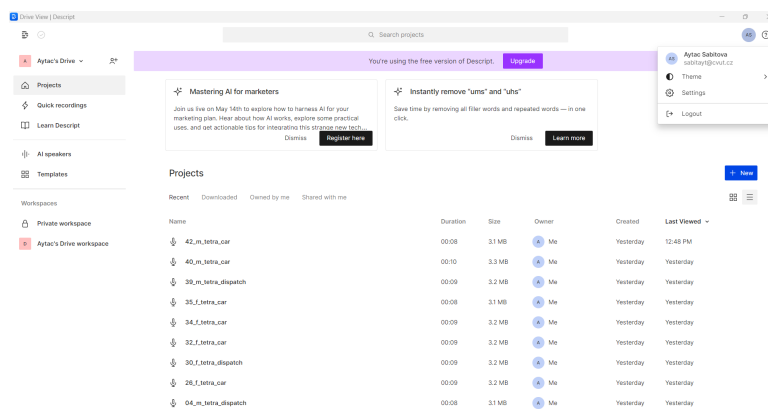


Figure 7.1: Descript.

7. Practical experiment

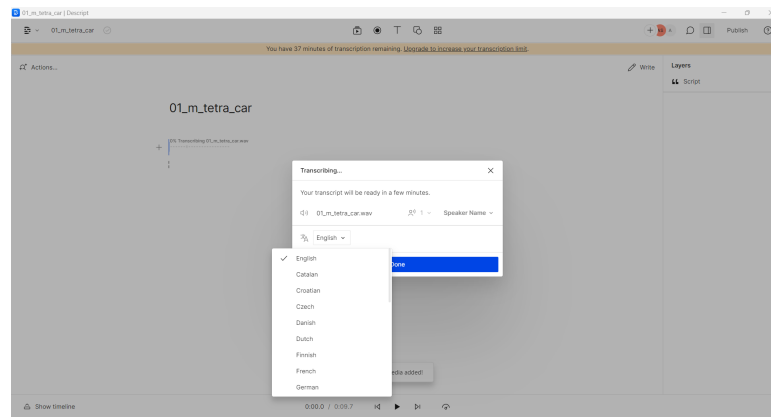


Figure 7.2: Descript.

The way Descript works, makes things look so easy, giving a relaxed working environment with good accuracy.

■ Speech-to-text on Whisper AI(local)

The next tool is Whisper AI(local). To begin with, first, you need to navigate to the folder that has all of your audio files. In File Explorer, click into the address field and then type **cmd** and then press enter. This opens up the command prompt, and now we're in the same directory that all of our files are in. To run Whisper, simply type in **whisper**, and then type in the file name and hit enter. By default, this will use the small model. But you can change model types. We can see that it automatically detects the language used in the field, and successfully identifies it. To minimize the command prompt, you can specify the language.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\99450\Desktop\whisper_raw\whisper > whisper "26_f_tetra_car.wav"
C:\Users\99450\AppData\Local\Programs\Python\Python39\lib\site-packages\whisper\transcribe.py:115: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warn(
warnings.warn("FP16 is not supported on CPU; using FP32 instead")
Detecting language using up to the first 30 seconds. Use '--language' to specify the language
[00:00.000 -> 00:00.400] 15, 58, 1, Simona Cyril.

C:\Users\99450\Desktop\whisper_raw\whisper > whisper "26_f_tetra_car.wav" --language Czech
C:\Users\99450\AppData\Local\Programs\Python\Python39\lib\site-packages\whisper\transcribe.py:115: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warn(
warnings.warn("FP16 is not supported on CPU; using FP32 instead")
[00:00.000 -> 00:00.400] 15, 58, 1, Simona Cyril.

C:\Users\99450\Desktop\whisper_raw>
```

Figure 7.3: Whisper AI(local).

■ Speech-to-text on Whisper AI(cloud)

Our next tool is Whisper AI(cloud). To begin with, we need to navigate our Google Drive, and then open Google Colaboratory. After opening our file,


```
#include <vosk_api.h>
#include <stdio.h>

int main() {
    FILE *wavin;
    char buf[3200];
    int nread, final;

    VoskModel *model = vosk_model_new("model");
    VoskRecognizer *recognizer = vosk_recognizer_new(model, 16000.0);

    wavin = fopen("test.wav", "rb");
    fseek(wavin, 44, SEEK_SET);
    while (!feof(wavin)) {
        nread = fread(buf, 1, sizeof(buf), wavin);
        final = vosk_recognizer_accept_waveform(recognizer, buf, nread);
        if (final) {
            printf("%s\n", vosk_recognizer_result(recognizer));
        } else {
            printf("%s\n", vosk_recognizer_partial_result(recognizer));
        }
    }
    printf("%s\n", vosk_recognizer_final_result(recognizer));

    vosk_recognizer_free(recognizer);
    vosk_model_free(model);
    fclose(wavin);
    return 0;
}
```

Figure 7.5: Vosk code

7.2 Experimental Results

This section presents our results after experiments. For experiments, we had four pairs of setups and in the following subsections, we will discuss the results from these experiments:

7.2.1 First Experiment

In our first experimental setup, we used Descript as a Speech-to-text tool, along with 40 audio sets which are preprocessed by four different audio signal processing tools. Descript has both audio features and speech-to-text itself, so we wanted to see how it performs with other audio tools. The other audio signal processing tools used on this set-up were iZotope RX 10, Audacity, and

Audacity x RX 10. Each tool preprocessed 10 sets from the total set of 40.

First Experiment Setup
Descript x Descript
Descript x iZotope RX 10
Descript x Audacity
Descript x Audacity x RX 10

Table 7.1: First Experiment Setup

First, we started with Descript itself, followed by iZotope RX 10, Audacity, and Audacity x RX10. We played audio files one by one and after getting the transcription we calculated the WER score. Since we already know the correct translations, calculating WER results was straightforward.

Here are the results after the experiment:

Users	Descript	iZotope RX 10	Audacity	Audacity x RX 10
Male 1	0 (0.0%)	0.357 (35.7%)	0 (0.0%)	0 (0.0%)
Male 2	0 (0.0%)	1 (100.0%)	0 (0.0%)	0 (0.0%)
Female 1	0 (0.0%)	1 (100.0%)	0 (0.0%)	0 (0.0%)
Female 2	0 (0.0%)	0.182 (18.2%)	0 (0.0%)	0 (0.0%)
Female 3	0 (0.0%)	0.4 (40.0%)	0 (0.0%)	0 (0.0%)
Female 4	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Female 5	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Male 3	0.2 (20.0%)	0 (0.0%)	0.2 (20.0%)	0.2 (20.0%)
Male 4	0 (0.0%)	0.714 (71.4%)	0 (0.0%)	0 (0.0%)
Male 5	0.667 (66.7%)	0 (0.0%)	1 (100.0%)	0.571 (57.1%)

Table 7.2: WER results for the first experiment

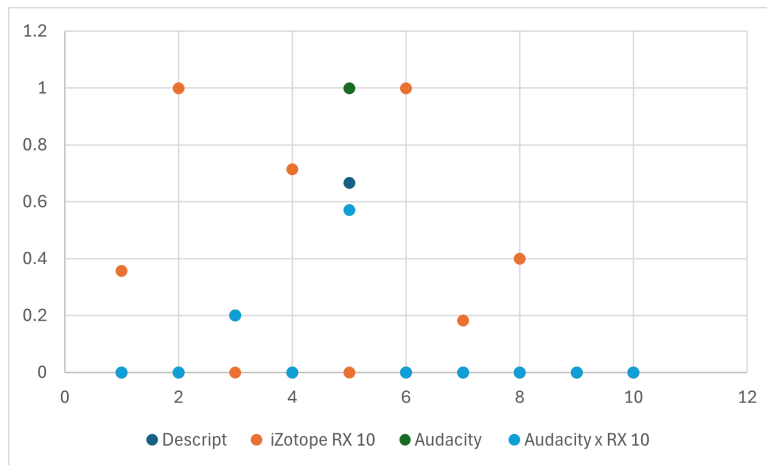


Figure 7.6: Comparison chart of Descript.

After calculating the WER score, we can analyze the scores. With different

setups, we got different scores. From WER scores, Describe itself gives an average of 0.0867 as the overall WER score with the male WER score at 0.1734 and the female WER score at 0. When we used audio that was preprocessed on iZotope RX 10, we got 0.3653 as the average overall WER score with the male WER score at 0.4142 and the female WER score at 0.3164. When we used audio from Audacity, we got 0.12 as the average overall WER score with the male WER score at 0.24 and the female WER score at 0. Finally, when we used Audacity x Rx 10, which produced the lowest, we got 0.0771 as the average overall WER score with the male WER score at 0.1542 and the female WER score at 0.

Users	Describe	iZotope RX 10	Audacity	Audacity x RX 10
Overall WER	0.0867	0.3653	0.12	0.0771
Male WER	0.1734	0.4142	0.24	0.1542
Female WER	0	0.3164	0	0

Table 7.3: WER results for the first experiment

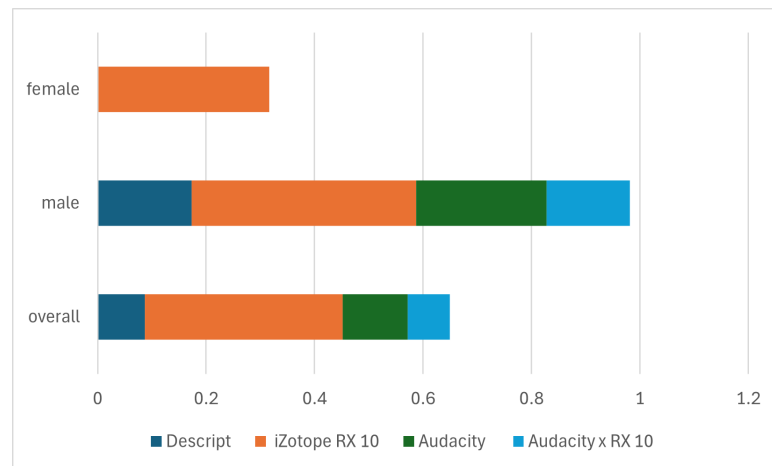


Figure 7.7: Comparison WER scores of Describe.

7.2.2 Second Experiment

In our second experimental setup, we used Whisper AI (local) as a Speech-to-text tool, along with 40 audio sets which are preprocessed by four different audio signal processing tools. The audio signal processing tools used in this set-up were Describe, iZotope RX 10, Audacity, and Audacity x RX 10. Each tool preprocessed 10 sets from the total set of 40.

Second Experiment Setup
Whisper AI (local) x Descript
Whisper AI (local) x iZotope RX 10
Whisper AI (local) x Audacity
Whisper AI (local) x Audacity x RX 10

Table 7.4: Second Experiment Setup

Before starting experiments with preprocessed data, we first experimented with a raw dataset to see how it performed. This helped us compare its performance with both raw and preprocessed data and see if any improvements were made.

First, let's see how Whisper AI(local) reacts to raw data:

Users	Raw data
Male 1	0.308 (30.8%)
Male 2	0.667 (66.7%)
Female 1	0 (0.0%)
Female 2	0.273 (27.3%)
Female 3	0.6 (60.0%)
Female 4	0 (0.0%)
Female 5	0 (0.0%)
Male 3	0.8 (80.0%)
Male 4	0.857 (85.7%)
Male 5	1.667 (166.7%)

Table 7.5: WER of Whisper AI(local) x raw data

For the experiment, we started with Descript, followed by iZotope RX 10, Audacity, and Audacity x RX10. We played audio files one by one and after getting the transcription we calculated the WER score. Since we already know the correct translations, calculating WER results was straightforward.

Here are the results after the experiment:

Users	Descript	iZotope RX 10	Audacity	Audacity x RX 10
Male 1	0.714 (71.4%)	1.071 (107.1%)	0.644 (64.3%)	0.286 (28.6%)
Male 2	0.5 (50.0%)	0.75 (75.0%)	0.25 (25.0%)	0.667 (66.7%)
Female 1	2 (200.0%)	2.4 (240.0%)	0 (0.0%)	0 (0.0%)
Female 2	0.364 (36.4%)	0.455 (45.5%)	0.273 (27.3%)	0.273 (27.3%)
Female 3	1 (100.0%)	0.6 (60.0%)	1.2 (120.0%)	0.6 (60.0%)
Female 4	1.571 (157.1%)	0 (0.0%)	0.143 (14.3%)	0 (0.0%)
Female 5	0 (0.0%)	0.7 (70.0%)	0.625 (62.5%)	0 (0.0%)
Male 3	1 (100.0%)	0.9 (90.0%)	0.8 (80.0%)	0.8 (80.0%)
Male 4	0.714 (71.4%)	0.857 (85.7%)	0.857 (85.7%)	1.429 (142.9%)
Male 5	1 (100.0%)	1 (100.0%)	1.143 (114.3%)	1.667 (166.7%)

Table 7.6: WER results for the second experiment

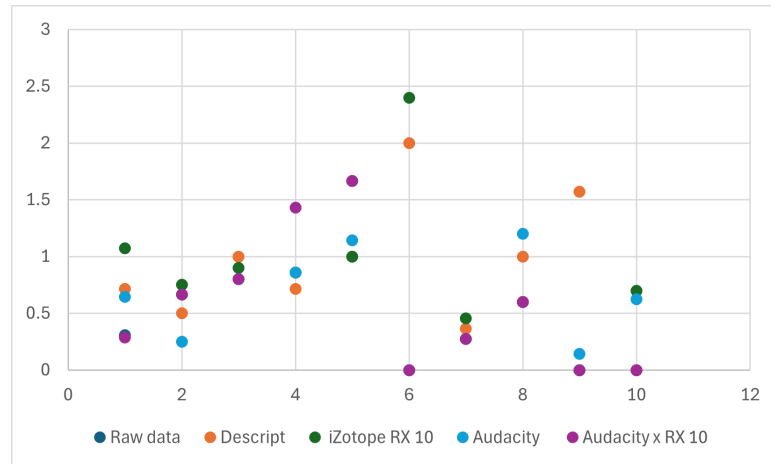


Figure 7.8: Comparison chart of Whisper(local).

After calculating the WER score, we can analyze the scores. With different setups, we got different scores. With raw data, we are getting 0.5172 as the average WER score with the male WER score at 0.8598 and the female WER score at 0.1746. From WER scores, Descript gives an average of 0.8863 as the overall WER score with the male WER score at 0.7856 and the female WER score at 0.987. When we used audio that was preprocessed on iZotope RX 10, we got 0.8733 as the average overall WER score with the male WER score at 0.9156 and the female WER score at 0.831. When we used audio from Audacity, we got 0.5935 as the average overall WER score with the male WER score at 0.7388 and the female WER score at 0.4482. Finally, when we used Audacity x Rx 10, we got 0.5722 the average overall WER score with the male WER score at 0.9698 and the female WER score at 0.1746.

Users	Raw data	Descript	iZotope RX 10	Audacity	Audacity x RX 10
Overall WER	0.5172	0.8863	0.8733	0.5935	0.5722
Male WER	0.8598	0.7856	0.9156	0.7388	0.9698
Female WER	0.1746	0.987	0.831	0.4482	0.1746

Table 7.7: WER results for the second experiment

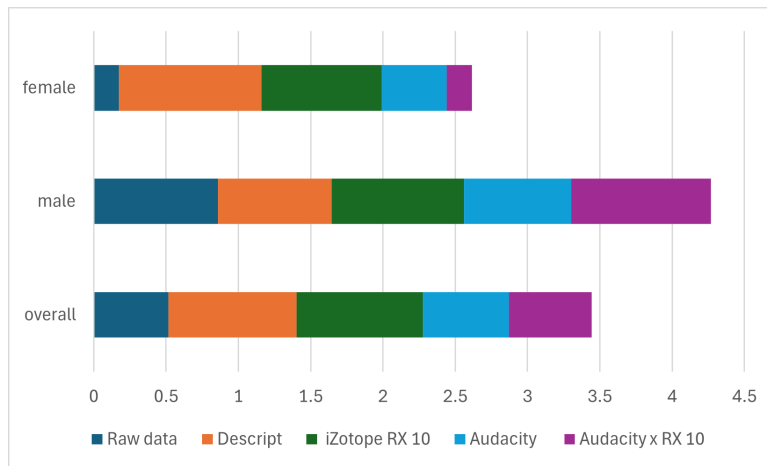


Figure 7.9: Comparison WER scores of Whisper(local).

7.2.3 Third Experiment

In our third experimental setup, we used Whisper AI (cloud) as a Speech-to-text tool, along with 40 audio sets which are preprocessed by four different audio signal processing tools. The audio signal processing tools used in this set-up were Descript, iZotope RX 10, Audacity, and Audacity x RX 10. Each tool preprocessed 10 sets from the total set of 40.

Third Experiment Setup
Whisper AI (cloud) x Descript
Whisper AI (cloud) x iZotope RX 10
Whisper AI (cloud) x Audacity
Whisper AI (cloud) x Audacity x RX 10

Table 7.8: Third Experiment Setup

Before starting experiments with preprocessed data, we first experimented with a raw dataset to see how it performed. This helped us compare its performance with both raw and preprocessed data and see if any improvements were made.

First, let's see how Whisper AI (cloud) reacts to raw data:

Users	Raw data
Male 1	0.143 (14.3%)
Male 2	0 (0.0%)
Female 1	0 (0.0%)
Female 2	0.182 (18.2%)
Female 3	0.2 (20.0%)
Female 4	0 (0.0%)
Female 5	0 (0.0%)
Male 3	0.1 (10.0%)
Male 4	0.429 (42.9%)
Male 5	1 (100.0%)

Table 7.9: WER of Whisper AI(cloud) x raw data

For the experiment, we started with Descript, followed by iZotope RX 10, Audacity, and Audacity x RX10. We played audio files one by one and after getting the transcription we calculated the WER score. Since we already know the correct translations, calculating WER results was straightforward.

Here are the results after the experiment:

Users	Descript	iZotope RX 10	Audacity	Audacity x RX 10
Male 1	0.571 (57.1%)	0.429 (42.9%)	0.071 (7.1%)	0.071 (7.1%)
Male 2	0.5 (50.0%)	1 (100.0%)	0 (0.0%)	0 (0.0%)
Female 1	1.8 (180.0%)	2.8 (280.0%)	0 (0.0%)	0 (0.0%)
Female 2	0.091 (9.1%)	0.364 (36.4%)	0.091 (9.1%)	0.182 (18.2%)
Female 3	1.2 (120.0%)	0 (0.0%)	0.2 (20.0%)	0 (0.0%)
Female 4	1.429 (142.9%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Female 5	0.125 (12.5%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Male 3	0.9 (90.0%)	0.333 (33.3%)	0.333 (33.3%)	0.111 (11.1%)
Male 4	0 (0.0%)	1.286 (128.6%)	0.571 (57.1%)	1 (100.0%)
Male 5	1.571 (157.1%)	0.857 (85.7%)	0.571 (57.1%)	0.714 (71.4%)

Table 7.10: WER results for the third experiment

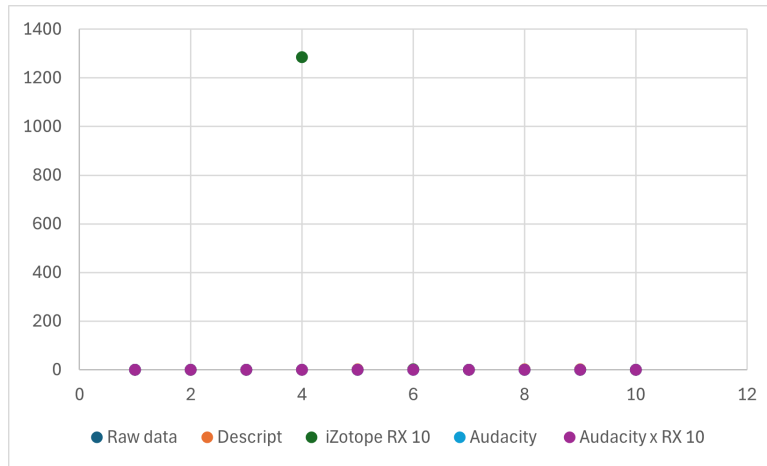


Figure 7.10: Comparison chart of Whisper(cloud).

After calculating the WER score, we can analyze the scores. With different setups, we got different scores. With raw data, we are getting 0.2054 as the average WER score with the male WER score at 0.3344 and the female WER score at 0.0764. From WER scores, Descript gives an average of 0.8187 as the overall WER score with the male WER score at 0.7084 and the female WER score at 0.929. When we used audio that was preprocessed on iZotope RX 10, we got 1.297181 as the average overall WER score with the male WER score at 0.781 and the female WER score at 0.6328. When we used audio from Audacity, we got 0.1837 as the average overall WER score with the male WER score at 0.571 and the female WER score at 0.0582. Finally, when we used Audacity x Rx 10, we got 0.2078 the average overall WER score with the male WER score at 0.3792 and the female WER score at 0.0364.

Users	Raw data	Descript	iZotope RX 10	Audacity	Audacity x RX 10
Overall WER	0.2054	0.8187	1.2971	0.1837	0.2078
Male WER	0.3344	0.7084	0.781	0.571	0.3792
Female WER	0.0764	0.929	0.6328	0.0582	0.0364

Table 7.11: WER results for the third experiment

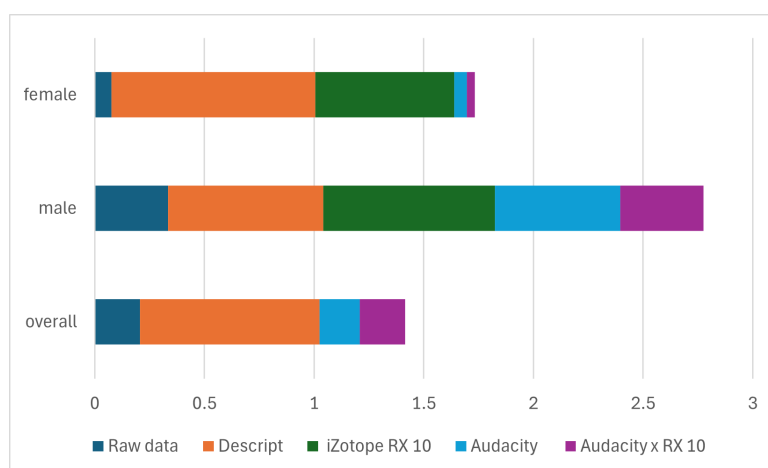


Figure 7.11: Comparison WER scores of Whisper(cloud).

7.2.4 Fourth Experiment

Finally, in our last, fourth experimental setup, we used Vosk as a Speech-to-text tool, along with 40 audio sets which are preprocessed by four different audio signal processing tools. The audio signal processing tools used in this set-up were Descript, iZotope RX 10, Audacity, and Audacity x RX 10. Each tool preprocessed 10 sets from the total set of 40.

Fourth Experiment Setup
Vosk x Descript
Vosk x iZotope RX 10
Vosk x Audacity
Vosk x Audacity x RX 10

Table 7.12: Fourth Experiment Setup

Before starting experiments with preprocessed data, we first experimented with a raw dataset to see how it performed. This helped us compare its performance with both raw and preprocessed data and see if any improvements were made.

First, let's see how Vosk reacts to raw data:

Users	Raw data
Male 1	0.4 (40.0%)
Male 2	0.767 (76.7%)
Female 1	0.091 (9.1%)
Female 2	0.363 (36.3%)
Female 3	0.8 (80.0%)
Female 4	0.091 (9.1%)
Female 5	0.091 (9.1%)
Male 3	0.75 (75.0%)
Male 4	0.857 (85.7%)
Male 5	1.867 (186.7%)

Table 7.13: WER of Vosk x raw data

For the experiment, we started with Descript, followed by iZotope RX 10, Audacity, and Audacity x RX10. We played audio files one by one and after getting the transcription we calculated the WER score. Since we already know the correct translations, calculating WER results was straightforward.

Here are the results after the experiment:

Users	Descript	iZotope RX 10	Audacity	Audacity x RX 10
Male 1	0.7 (70.0%)	1 (100.0%)	0.9 (90.0%)	0.386 (38.6%)
Male 2	0.6 (60.0%)	0.65 (65.0%)	0.65 (65.0%)	0.677 (67.7%)
Female 1	1 (100.0%)	1.4 (140.0%)	0.091 (9.1%)	0.091 (9.1%)
Female 2	0.664 (66.4%)	0.45 (45.0%)	0.45 (45.0%)	0.35 (35.0%)
Female 3	1 (100.0%)	0.9 (90.0%)	1.2 (120.0%)	0.9 (90.0%)
Female 4	1.6 (160.0%)	0.714 (71.4%)	0.143 (14.3%)	0.091 (9.1%)
Female 5	0.182 (18.2%)	0.65 (65.0%)	0.6 (60.0%)	0.091 (9.1%)
Male 3	1 (100.0%)	0.9 (90.0%)	0.8 (80.0%)	0.8 (80.0%)
Male 4	0.617 (61.7%)	1 (100.0%)	0.657 (65.7%)	1.72 (172.0%)
Male 5	1.5 (150.0%)	1.5 (150.0%)	1.547 (154.7%)	1 (100.0%)

Table 7.14: WER results for the fourth experiment

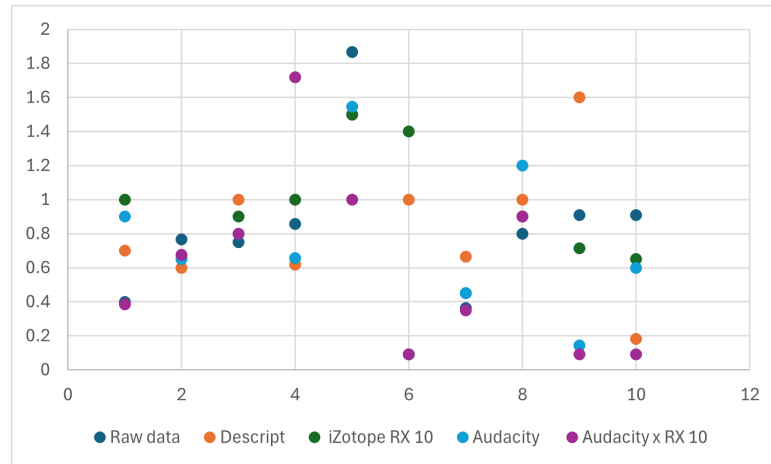


Figure 7.12: Comparison chart of Vosk.

After calculating the WER score, we can analyze the scores. With different setups, we got different scores. With raw data, we are getting 0.7715 as the average WER score with the male WER score at 0.9282 and the female WER score at 0.6148. From WER scores, Descript gives an average of 0.8863 as the overall WER score with the male WER score at 0.8834 and the female WER score at 0.8892. When we used audio that was preprocessed on iZotope RX 10, we got 0.9164 as the average overall WER score with the male WER score at 1.01 and the female WER score at 0.8228. When we used audio from Audacity, we got 0.7038 as the average overall WER score with the male WER score at 0.9108 and the female WER score at 0.4968. Finally, when we used Audacity x Rx 10, we got 0.6106 as the average overall WER score with the male WER score at 0.9166 and the female WER score at 0.3046.

Users	Raw data	Descript	iZotope RX 10	Audacity	Audacity x RX 10
Overall WER	0.7715	0.8863	0.9164	0.7038	0.6106
Male WER	0.9282	0.8834	1.01	0.9108	0.9166
Female WER	0.6148	0.8892	0.8228	0.4968	0.3046

Table 7.15: WER results for the fourth experiment

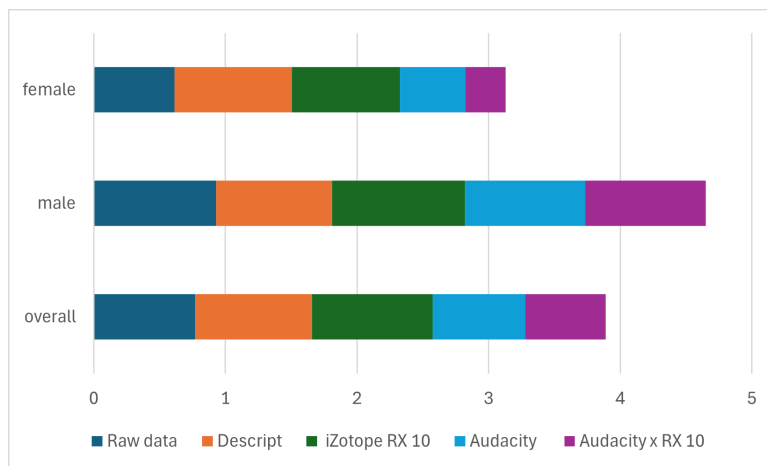


Figure 7.13: Comparison WER scores of Vosk.

7.3 Analysis of ASR system performance and key findings

After using different Automatic Speech Recognition (ASR) systems, and finishing our experiments, we will discuss the performance of these systems.

Performance of the ASR tools

Starting with Descript, even if it's more known for its audio editing features, as speech-to-text tool it justified its work. If we check overall WER results, it's nearly the perfect tool for speech-to-text. WER scores follow as, with **Descript** itself giving an average of 0.0867 as the overall WER score with the male WER score at 0.1734 and the female WER score at 0, with **iZotope RX 10**, we got 0.3653 as the average overall WER score with the male WER score at 0.4142 and the female WER score at 0.3164, with **Audacity**, we got 0.12 as the average overall WER score with the male WER score at 0.24 and the female WER score at 0, and with **Audacity x Rx 10**, we got 0.0771 as the average overall WER score with the male WER score at 0.1542 and the female WER score at 0. It worked with other audio tools fairly enough except slightly worse with iZotope RX 10. Despite this, the tool works amazingly with both male and female audio.

For our next speech-to-text tool, we chose Whisper AI, both in local and cloud environments. Starting with the local environment, was fairly good as well. But it didn't make any huge difference between raw data and preprocessed data. Numbers were quite similar between raw data and preprocessed data. WER scores follow as, with **raw data**, we are getting 0.5172 as the average WER score with the male WER score at 0.8598 and the female WER score at 0.1746, with **Descript** giving an average of 0.8863 as the overall WER score with the male WER score at 0.7856 and the female WER score at 0.987, with **iZotope RX 10**, we got 0.8733 as the average overall WER score with the male WER score at 0.9156 and the female WER

score at 0.831, with **Audacity**, we got 0.5935 as the average overall WER score with the male WER score at 0.7388 and the female WER score at 0.4482 and with **Audacity x Rx 10**, we got 0.5722 the average overall WER score with the male WER score at 0.9698 and the female WER score at 0.1746. One issue that we had with the local environment is, its automatic language detection, it detected the wrong language. With English audio files there wasn't any problem but since we used Czech audio files, it started to detect the wrong language. It showed Polish or Albanian language as the detected language if we don't specify the language on command. Overall speaking, the way it was quick and had decent accuracy was quite good.

On the other hand, using it in a cloud environment was much more seamless. Specially, if you do not want to install anything on your local computer. It has fairly good translation time as well, the only issue could be that you need to download your files after transcription since it deletes them automatically after you end your runtime. WER scores follow as, with **raw data**, we are getting 0.2054 as the average WER score with the male WER score at 0.3344 and the female WER score at 0.0764, with **Descript** giving an average of 0.8187 as the overall WER score with the male WER score at 0.7084 and the female WER score at 0.929, with **iZotope RX 10**, we got 1.297181 as the average overall WER score with the male WER score at 0.781 and the female WER score at 0.6328, with **Audacity**, we got 0.1837 as the average overall WER score with the male WER score at 0.571 and the female WER score at 0.0582 and with **Audacity x Rx 10**, we got 0.2078 the average overall WER score with the male WER score at 0.3792 and the female WER score at 0.0364. Looking at WER results, interestingly, it seems like on the cloud platform, it works better with raw data compared to preprocessed data. But overall speaking, in both environments, the tool worked better with female than male audio.

Finally, the last tool, we had Vosk. Overall, speaking, it didn't any huge difference in WER scores between raw data and preprocessed data. However, the problem with this tool is that it took more time than other speech-to-text tools. Another issue is, that in the middle of transcription, it will give the wrong words, if sentences get longer, it gives more different words. WER scores follow as, with **raw data**, we are getting 0.7715 as the average WER score with the male WER score at 0.9282 and the female WER score at 0.6148, with **Descript** giving an average of 0.8863 as the overall WER score with the male WER score at 0.8834 and the female WER score at 0.8892, with **iZotope RX 10**, we got 0.9164 as the average overall WER score with the male WER score at 1.01 and the female WER score at 0.8228, with **Audacity**, we got 0.7038 as the average overall WER score with the male WER score at 0.9108 and the female WER score at 0.4968 and with **Audacity x Rx 10**, we got 0.6106 as the average overall WER score with the male WER score at 0.9166 and the female WER score at 0.3046. Overall speaking, the tool worked better with female than male audio.

■ Fairness in WER

For the calculation of our ASR performance results, we used the word

error rate (WER). Calculation is done by counting the number of words that need to be substituted (S), deleted (D), and inserted (I) to transition from a ground-truth transcription to the output of an ASR system. This count is then divided by the total number of words in the ground-truth transcription (N). It's a known fact that WER treats each discrepancy between the ground-truth transcription and the ASR output equally.

However, after experimental results, it was clear that not all Automatic Speech Recognition (ASR) errors are equal. Word Error Rate (WER) results don't consistently align with human judgment or performance in downstream tasks like information retrieval, natural language understanding, or named entity recognition.

We can see a reference to this talk on 4.3 where we discussed the sentence "I love you." If one ASR system outputs "I loathe you," and another predicts "I love you," both cases result in a WER of 0.33. However, the severity of the errors differs significantly. While "luv" is a minor deviation from the ground truth "love," "loathe" completely changes the meaning of the sentence.

Same with other words that can be understandable for humans but could be considered as errors on Automatic Speech Recognition (ASR) tools and algorithms.

Despite these challenges, overall, all of these tools did their job pretty well, even without preprocessed data they can give good accuracy speech-to-text transcription. After experiments, it is clear to us that Automatic Speech Recognition (ASR) has bias regarding gender. It looks like it works better with female audio. However, we need to consider factors such as speech, speaker, channel, and environmental conditions. Meanwhile, Word Error Rate (WER) results do not always align with human judgment or performance.

Even though the number of audio samples used in this experiment was relatively small, it didn't affect our purpose and served us to understand the core of ASR and identify any potential improvements after testing. However, we acknowledge that testing on larger databases is essential for a comprehensive understanding of ASR, and it remains one of our future research objectives.

7.4 Future Research Directions in ASR

Over the years, the field of ASR proceeded from its early stages and quickly grew the number of practical applications and commercial markets. Despite its achievements, ASR remains far from being a solved problem. As in the past, we expected further research and development, there's still much room for further improvement.

Moore's Law, which predicts the doubling of computational power every 12 to 18 months, has played a significant role in ASR development. This has allowed researchers to run increasingly complex algorithms quickly, leading

to significant progress. Access to common speech corpora for training and evaluation has also been crucial in enhancing ASR systems' capabilities.

However, speech is highly variable and influenced by many factors such as speech, speaker, channel, and environmental conditions. ASR systems often struggle when faced with signals that differ from their training data, even slightly. A key area of research focuses on improving the robustness of ASR systems against such variability and shifts in acoustic environments, noise sources, speaker characteristics, and language characteristics.[23]

Future research can aim to develop systems that can adapt to diverse conditions, such as different speaking styles, accents, emotional states, and language variations. New techniques and architectures are being explored to automatically adjust to changing conditions in real-time, enabling accurate transcription in various environments, from formal presentations to casual conversations.

It became clear that not all Automatic Speech Recognition (ASR) errors are equal and Word Error Rate (WER) results don't consistently align with human judgment. In order to have fairness in ASR results, future research can aim to develop metrics that align with human judgment and performance in downstream tasks. Considering information like retrieval, natural language understanding, or named entity recognition.

Addressing this challenge requires collaboration across disciplines such as natural language processing, information retrieval, and cognitive science. The ultimate goal will be to make ASR systems more reliable and useful in a wide range of applications, facilitating communication and interaction in diverse settings.[23]

Chapter 8

Conclusion

The purpose of this thesis assignment, which involved investigating Automatic Speech Recognition (ASR) through experiments, has been effectively achieved, with all stated goals met.

Over the years speech technologies improved significantly, especially Automatic Speech Recognition (ASR). Automatic Speech Recognition (ASR) is a vital bridge in human-machine interactions. It involves several areas from military communications to emergency response systems. Even though there are significant improvements, Automatic Speech Recognition (ASR) still faces challenges. The performance of the system can depend on environment, gender, and language.

In this thesis, we aimed to enhance the success rate of speech radio channel content recognition. By reviewing state-of-the-art methods, presenting our approach, and comparing results, we contribute to a deeper understanding of speech, digital signal processing, and Automatic Speech Recognition (ASR) systems. Our research underscores the importance of optimizing these systems for practical applications.

For our experiments, we had four different setups. For each setup, we focused on one Automatic Speech Recognition (ASR) system and 40 audio sets, preprocessed by four different audio signal processing tools. The Automatic Speech Recognition (ASR) systems we used were Descript, Whisper AI (in both local and cloud environments), and Vosk. The audio signal processing tools we used were Descript, iZotope RX 10, Audacity, and Audacity x RX 10. For audio, we used 10 sets from 5 different male and female speakers. Each tool preprocessed 10 sets from the total set of 40.

If we look closely, Descript demonstrates near-perfect accuracy. The WER scores follow as: Descript itself gave an average of 0.0867, iZotope RX 10 gave 0.3653, Audacity gave 0.12 and Audacity x Rx 10 gave 0.0771.

For gender WER scores, Descript itself gave an average male WER of 0.1734 and female WER of 0, iZotope RX 10 gave an average male WER of 0.4142 and female WER of 0.3164, Audacity gave an average male WER of 0.24 and female WER of 0, and Audacity x RX 10 gave an average male WER of 0.1542 and female WER of 0. Comparing male and female WER scores, Descript works better with female voices compared to male.

For our next speech-to-text tool, we chose Whisper AI, both in local

there wasn't a huge difference between raw data and data reprocessed with Audacity and Audacity x RX 10, but slightly worse WER scores with Descript and iZotope RX 10. Descript achieved near-perfect accuracy but had slightly worse WER scores with iZotope RX. In addition, if we compare male and female WER scores, we can see that on every setup, ASR has a bias on female voices and performing better with them.

Even with these challenges, all tools provide reliable speech-to-text transcription. However, it's important to note that, the speaker has a role in the performance of these systems too, such as speaking style, emotional state, or environment, which can affect results. Additionally, while WER is commonly used for calculation, it may not always align with human judgment or performance. Our experiment used a relatively small number of audio samples, but it served our purpose of understanding ASR and identifying potential improvements, and how the performance of the system can depend on environment, gender, and language. However, we acknowledge that testing on larger databases is essential for a comprehensive understanding of ASR, and it remains one of our future research objectives.

With these findings, ongoing research is needed to further improve ASR systems. Despite the acceleration in ASR development due to Moore's Law, challenges remain due to the influence of factors like environment and speaker characteristics on speech. Future efforts should focus on improving system robustness, adapting to diverse conditions, and enabling real-time adjustments for accurate transcription across different contexts. For fairness of ASR systems proper metrics considering relevant information should be developed to align ASR systems with human judgment and performance in downstream tasks.

Addressing these challenges requires collaboration and innovative approaches. To make Automatic Speech Recognition (ASR) systems more reliable and adaptable, we can turn these systems into more effective communication and interaction in various settings.



Bibliography

- [1] STAN Z. LI, Anil J.: Encyclopedia of Biometrics.
- [2] ATMAJA, Bagus T.: The Physiology, Mechanism, and Nonlinearities of Hearing. (2021)
- [3] C. STARR, C. E. ; STARR, L.: Biology: A human emphasis, ser. Cengage Advantage Books. In: *Cengage Learning; 8 edition, isbn: 0538757027* (2010)
- [4] FLETCHER, H. ; MUNSO, W.: Loudness, its definition, measurement and calculation. In: *Journal of the Acoustic Society of America* volume 5 (1933)
- [5] LI TAN, Jean J.: Digital Signal Processing: Fundamentals and Applications. (2019)
- [6] HUANG LU, Wang Haodong Li Jin Ma Xuejiao Zhang C. Yuan Xiaoyu X. Yuan Xiaoyu: Research on Application of Digital Signal Processing Technology in Communication. In: *Qinghai Normal University, Xi'ning, Qinghai, 810000, China*
- [7] LAWRENCE R. RABINER, Ronald W. S.: Introduction to Digital Speech Processing. In: *Springer, ISBN 978-1-4471-5779-3* volume 1 (2007)
- [8] ALIM, Sabur A. ; RASHID, Nahrul Khair A.: Some Commonly Used Speech Feature Extraction Algorithms.
- [9] RYAN WHETTEN, Casey K.: Evaluating and Improving Automatic Speech Recognition using Severity.
- [10] RANSOME, J. F. ; RITTINGHOUSEA, J.: Voice over internet protocol. In: *Digital Press* (2005)
- [11] GERHARD, David: Pitch Extraction and Fundamental Frequency: History and Current Techniques. In: *Technical Report TR-CS 2003-06* (2003)
- [12] UHLIR, J.: echnologie hlasovych komunikac. In: *CVUT Praha, 2007, isbn: 978-80-01-03888-8* (2007)

- [30] ZHANG, Park D. S. Han W. Qin J. Gulati-A. Shor J. Jansen A. Xu Y. Huang Y. Wang S. et a. Y.: Exploring the frontier of large-scale semi-supervised learning for automatic speech.
- [31] JINYU LI, Reinhold Häb-Umbach Yifan G. Li Deng D. Li Deng: Robust Automatic Speech Recognition. In: *ISBN: 9780128023983*
- [32] DEMIR, Dogan M U. Cemgil-A T. Saraclar M. C.: Catalog-based single-channel speech-music separation for automatic speech recognition. (2012)
- [33] MA, D: Multitask-Based Joint Learning Approach To Robust ASR For Radio Communication Speech. (2021)
- [34] CATELLIER, Andrew ; VORAN, Stephen: Wideband Audio Waveform Evaluation Networks: Efficient, Accurate Estimation of Speech Qualities.
- [35] ASMA TRABELSIA, Yassine Aajaounb-Severine S. Sebastien Waricheta W. Sebastien Waricheta: Evaluation of the efficiency of state-of-the-art Speech Recognition engines. In: *26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*
- [36] JUAN ZULUAGA-GOMEZ, Amrutha Prasad Petr Motlicek Driss Khalil Srikanth Madikeri Allan Tart Igor Szoke Vincent Lenders Mickael Rigault Khalid C. Iuliia Nigmatulina N. Iuliia Nigmatulina: Lessons Learned in ATCO2: 5000 hours of Air Traffic Control Communications for Robust Automatic Speech Recognition and Understanding. (2023)
- [37] JUNEJA, Amit: A comparison of automatic and human speech recognition in null grammar. In: *J. Acoust. Soc. Am. 131, EL256-EL261* (2012)
- [38] MALGORZATA ANNA ULASIK, Fabian Germann Esin Gedik Fernando Benites Mark C. Manuela Hurlimann H. Manuela Hurlimann: CEASR: A Corpus for Evaluating Automatic Speech Recognition. In: *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), Marseille* (2020), Mai
- [39] JUAN ZULUAGA-GOMEZ, Igor Szöke Alexander Blatt Petr Motlicek Martin Kocour Mickael Rigault Khalid Choukri Amrutha Prasad Seyyed Saeed Sarfjoo Iuliia Nigmatulina Claudia Cevenini Pavel Kolcárek Allan Tart Jan Cernocký Dietrich K. Karel Veselý V. Karel Veselý: ATCO2 corpus A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications. (2023)
- [40] SALIHA BENKERZAZ, Abdeslam D. Youssef Elmir E. Youssef Elmir: A Study on Automatic Speech Recognition. In: *DOI: 10.6025/jitr/2019/10/3/77-85* (2019)

- [52] MAHMOUD GABER, Ahmed O. Gloria Corpas Pastor P. Gloria Corpas Pastor: Speech-to-Text Technology as a Documentation Tool for Interpreters: a new approach to compiling an ad hoc corpus and extracting terminology from video-recorded speeches. (2020)
- [53] REGIS PIRES MAGALH~AES, Guilherme Sales Fernandes Livia Almada Cruz Matheus Xavier Sampaio Jose Antonio Fernandes de Macedo Ticianiana Linhares Coelho da S. Daniel Jean Rodrigues Vasconcelos V. Daniel Jean Rodrigues Vasconcelos: Evaluation of Automatic Speech Recognition Approaches. In: *Journal of Information and Data Management* Vol. 13, No. 3 (2022), S. Pages 366–377
- [54] OLIVIER GALIBERT, Juliette Kahn Sophie R. Mohamed Ameer Ben Jannet J. Mohamed Ameer Ben Jannet: Generating Task-Pertinent sorted Error Lists for Speech Recognition. In: *1LNE, F-78190 Trappes, France*
- [55] LAWRENCE R. RABINER, Ronald W. S.: Introduction to Digital Speech Processing. In: *ISBN: 978-1-60198-070-0* (2007)
- [56] GALIBERT, Olivier: Methodologies for the evaluation of Speaker Diarization and Automatic Speech Recognition in the presence of overlapping speech. In: *Laboratoire national de m´etrologie et d’essais, Trappes, France, INTERSPEECH* (2013)
- [57] SCHALLER, ROBERT R.: MOORE’S LAW: past, present, and future. In: *IEEE SPECTRUM* (1997)
- [58] HARTMUT HELMKE, Shruthi Shetty Hörður Arilíusson Teodor S. Simiganoschi Matthias Kleinert Oliver Ohneiser Heiko Ehr Juan Zuluaga-Gomez Pavel S. Karel Ondřej O. Karel Ondřej: Readback Error Detection by Automatic Speech Recognition and Understanding; Results of HAAWAI project for Isavia’s Enroute Airspace. (2022)
- [59] YOUSSEF OUALIL, Hartmut Helmke Anna Schmidt Dietrich K. Marc Schulder S. Marc Schulder: Real-Time Integration of Dynamic Context Information for Improving Automatic Speech Recognition. In: *INTER SPEECH* (2015)
- [60] M.A.ANUSUYA: Speech Recognition by Machine: A Review. In: *(IJCSIS) International Journal of Computer Science and Information Security* Vol. 6, No. 3 (2009)
- [61] SAHAR GHANNAY, Nathalie C. Yannick Esteve E. Yannick Esteve: Task specific sentence embeddings for ASR error detection. In: *Interspeech 2018, Hyderabad, India* (2018)
- [62] FITCH, W. T.: The Biology and Evolution of Speech: A Comparative Analysis. In: *University of Vienna, Vienna 1090, Austria*
- [63] HACKETT, Charles F.: The Origin of Speech. (1960)

- [64] The Past Present and Future of Speech Processing. In: *IEEE SIGNAL PROCESSING MAGAZINE 1053-5888/98/\$10.00C 1998IEEE* (1998), Mai
- [65] ANNA ESPOSITO, Nikolaos Avouris Ioannis Hatzilygeroudis (. Nikolaos G. Bourbakis B. Nikolaos G. Bourbakis: Verbal and Nonverbal Features of Human-Human and Human-Machine Interaction. In: *COST Action 2102 International Conference Patras, Greece* (2007)



Appendix A

Abbreviations list

ASR	Automatic Speech Recognition
STT	Speech-to-text
TTS	Text-to-speech
WER	Word Error Rate
HNR	harmonics-to-noise ratio
SIL	Sound intensity level
SPL	Sound pressure level
FFT	Fast Fourier Transform
PLP	Perceptual Linear Prediction
MFCC	Mel-Frequency Cepstral Coefficients
LP	Linear Predictive
DSP	Digital Signal Processing
ADC	Analog-to-digital
DS	Digital Signal
DAC	Digital-to-analog
ECG	Electrocardiograms
EEG	Electroencephalograms
DHH	Users who are deaf or hard of hearing
FIR	Finite Impulse Response filter
DCT	Discrete Cosine Transform
LPC	Linear Prediction Coding
LPCC	Linear prediction cepstral coefficients
LAR	Log Area Ratio

