VU UNIVERSITY AMSTERDAM | UNIVERSITEIT VAN AMSTERDAM

RESEARCH PROJECT

Overcoming Complex Speech Scenarios in Audio Cleaning for Voice-to-Text

June 19, 2023

Student:
Antria Panayiotou
Student nr: 2735005

Tutor:
dr. Anna Bon

1 Introduction

Imagine a world where every spoken word could be effortlessly transformed into an accurate written text, revolutionizing communication and accessibility across languages and cultures. The aforementioned accomplishment is facilitated by Automatic Speech Recognition (ASR) systems, which transform oral communication into textual representation. The swift progressions in ASR technology have propelled us towards an era where spoken language can be effortlessly converted into precise written text. The improvement of ASR systems and resolution of intricate speech scenarios have emerged as a central area of focus in research and development. The process of converting speech to text is accompanied by a number of obstacles, encompassing a variety of speech patterns and difficulties arising from environmental factors, background noise, and variations in pronunciation.

Nevertheless, when considering low-resource settings and indigenous languages with limited resources, the process of converting speech to text encounters various obstacles. The aforementioned data sources encompass modest data corpuses, data procured via mobile devices amidst environmental noise, and dialectal discrepancies arising from regional dissimilarities. The restricted accessibility of data poses a challenge to the training of models, whereas audio captured through mobile devices may exhibit noise and distortions. Accurate transcription of regional dialects necessitates language models that are capable of capturing subtle nuances. Mitigating these challenges requires novel methodologies for minimising noise interference, devising effective data gathering approaches, and adapting to diverse dialects. The present study investigates the difficulties encountered in the process of audio cleaning for the purpose of converting speech to text. The study places special emphasis on identifying strategies to alleviate these challenges and integrating them to tackle intricate speech scenarios.

The act of speaking is a fundamental means of human communication that facilitates the transfer of concepts, information, and affective states. The emergence of ASR systems has significantly transformed our capacity to transcribe spoken language into written text, thereby facilitating diverse applications such as transcription services, voice assistants, and other related domains. The precision of ASR is significantly contingent upon the excellence of the audio input, which can frequently be undermined by intricate speech situations. ASR is a multidisciplinary field that involves various areas of study, including linguistics, signal processing, machine learning, and human-computer interaction. The process entails unraveling the complex correlation between acoustic waves and linguistic expressions, analyzing the nuances of human verbal communication, and utilizing advanced technological tools to enhance the precision of automatic speech recognition.

The quest for precise transcription has been a driving force behind human progress throughout the course of history. Throughout history, there has been a persistent drive to establish a connection between oral and written forms of communication, ranging from the carving of symbols into stone

by ancient civilizations to the development of the printing press. Currently, we are at the forefront of a new era, utilizing the capabilities of machine learning and artificial intelligence to reveal the mysteries of spoken language and facilitate effortless conversion of voice-to-text.

The present study aims to examine the key determinants that exert a substantial impact on the accuracy of ASR. The study places a specific emphasis on the influence of extraneous noise, speaker diversity, contextual factors, and phonetic mispronunciations. The aforementioned factors present significant obstacles in attaining dependable conversion of speech to text. Furthermore, the present study investigates the benefits and compromises linked to the utilization of diverse amalgamations of audio cleansing methodologies in the preprocessing of audio data for machine learning algorithms utilized in ASR.

The objective of this study is to offer useful insights and practical strategies to improve the accuracy of ASR in real-world applications by conducting a comprehensive analysis of intricate speech scenarios. By means of thorough examination, systematic experimentation, and meticulous assessment of extant literature, this study makes a valuable contribution to the progress of ASR technology and provides a basis for enhanced voice-to-text conversion. Our objective is to narrow the divide between oral communication and written language, with the aim of unleashing the complete capabilities of automatic speech recognition systems. This will facilitate improved communication, accessibility, and innovative applications across a broad spectrum of domains, including transcription services and voice-activated technologies. With every progressive stride, we are approaching the realization of a seamless and precise voice-to-text transcription system.

The paper is structured as follows. In Chapter 1, we present an introduction that emphasizes the importance of audio cleaning in the context of voice-to-text conversion and provides a captivating overview of the research topic. Chapter 2 focuses on the research questions, delving into the primary factors that significantly affect the accuracy of ASR systems and discussing strategies to mitigate their impact. In Chapter 3, we conduct a comprehensive literature review, comprising two subchapters. The first subchapter offers background information on ASR systems, exploring their functioning, limitations, and challenges faced in complex speech scenarios. The second subchapter analyzes the primary factors affecting ASR accuracy, drawing from existing research and studies in the field. Chapter 4 details the methodology employed in this study, encompassing data collection, experimental setup, and evaluation metrics. In Chapter 5, we delve into audio enhancement techniques for ASR systems, examining the effectiveness of approaches such as noise reduction, speaker variability training sets, speaker normalization techniques, the integration of a variety of speakers, and language models. Chapter 6 presents the evaluation methodology, highlighting the experimental results and engaging in a thorough discussion and analysis. Chapter 7 offers a comprehensive examination of the results, discussing the advantages and trade-offs associated with the employed techniques. Finally, in Chapter 8, we conclude the paper by summarizing the key findings, reflecting on the learnings derived from this research, and proposing potential avenues for future improvements in audio cleaning techniques for voice-to-text conversion applications.

2 Research Questions

Despite being widely used, ASR technology still has a number of issues that could decrease its accuracy and dependability. The wide range of speech patterns and acoustic conditions is one of the biggest obstacles. The accuracy of ASR systems may be impacted by this variation, resulting in mistakes and misinterpretations. In this paper, we explore two crucial ASR-related Research Questions (RQs) in order to address these issues:

RQ1: *What are the primary factors that significantly affect the accuracy of automatic speech recognition, and what strategies can be employed to mitigate their impact?*

The significance of addressing this research inquiry lies in the fact that the precision of ASR systems is often influenced by diverse factors, including but not limited to background noise, speaker variability, accents, and other related factors. Through the identification of key factors that exert a substantial influence on the precision of ASR and the formulation of corresponding mitigation strategies, it is **possible to enhance the dependability and efficacy** of such systems.

In order to address the research inquiry, a systematic review of the extant literature will be undertaken to ascertain the principal factors that exert a significant impact on the precision of ASR. Subsequently, experiments will be carried out to evaluate the effects of aforementioned factors on the precision of ASR systems. The proposed experiments will entail the utilization of diverse datasets comprising of recordings featuring distinct accents, speaking styles, ambient noise, and other pertinent variables. Subsequently, the obtained outcomes will be scrutinized to discern the pivotal variables that exert a substantial influence on the precision of ASR and ascertain the optimal tactics to alleviate their repercussions.

Several strategies that can be investigated to mitigate the influence of the primary factors that impact ASR precision encompass methods for reducing noise, adapting to the speaker, expanding the vocabulary, and normalizing accents. The techniques under consideration will be applied to a consistent corpus of audio data in order to assess their efficacy in enhancing accuracy and to determine the optimal approaches for mitigating the influence of the primary factors that impact automated speech recognition.

RQ2: *What are the advantages and trade-offs of employing various combinations of audio cleaning techniques in the preparation of audio files for machine learning models used in automatic speech recognition? Which cleaning techniques are most effective in improving the final results of the automatic speech recognition system?*

The significance of this inquiry stems from the fact that the precision of ASR systems is contingent upon the caliber of the audio data utilized for the purpose of training the machine learning algorithms. Audio recordings are frequently subject to various types of interference, such as background noise, reverberation, and other forms of distortion, which may have an impact on the precision of ASR) systems. The implementation of diverse amalgamations of audio cleansing methodologies can **enhance the caliber of audio data utilized for the training** of ASR models, thereby **augmenting their precision**.

In order to address this research inquiry, a comprehensive examination of the current body of literature pertaining to audio cleansing methodologies employed in the preprocessing of audio data for machine learning algorithms in the domain of ASR will be conducted. Subsequently, we shall assess the benefits and compromises of diverse amalgamations of audio refinement methodologies through a sequence of trials utilizing distinct sets of data. The study will additionally examine the effects of various audio cleaning methodologies on the precision of ASR systems. The objective of our study is to ascertain the optimal amalgamations of audio cleansing methodologies that can enhance the caliber of audio documents utilized for the instruction of ASR models, thereby elevating their precision.

3 Literature Review

3.1 Background information

3.1.1 Historical Overview of ASR

ASR technology has a lengthy and extensive chronicle that encompasses numerous decades.The following graphic, Figure 2 illustrates the evolution of ASR from 1950 to 2020. The inception of ASR can be attributed to the initial years of the 1950s, during which scholars commenced investigating the feasibility of utilizing computers for speech recognition. Davis et al. (1952) reported that Bell Laboratories developed an ASR system in the 1950s that utilized a formant-based methodology to identify vowel sounds [1]. Initially, speech recognition systems were primarily designed to recognize numerical inputs rather than linguistic ones. A decade subsequent to its predecessor, IBM unveiled "Shoebox," a language processing system capable of comprehending and generating responses to a vocabulary of 16 English words.

In the subsequent decades, the technology of ASR underwent further advancements, as scholars delved into diverse methodologies for speech recognition. The field of ASR experienced a significant advancement during the 1970s, when the technique of Hidden Markov Models (HMMs) was introduced for speech recognition by Baker in 1975[2]. HMMs are a type of statistical model that

enables the modeling of temporal dynamics in speech. This represents a significant improvement over prior methods that treated individual speech sounds as distinct entities. The HMM employed a probabilistic approach to estimate the likelihood of the unidentified phonemes constituting lexemes, rather than relying solely on phonetic features and auditory cues.

During the 1980s and 1990s, advancements in ASR technology were made through the creation of novel algorithms and models. A significant development during this timeframe pertained to the utilization of neural networks in the domain of speech recognition, as noted by Bourlard and Morgan (1994)[3]. Neural networks are a machine learning model that is capable of recognizing patterns in data. Studies have demonstrated their efficacy in speech recognition, particularly when compared to HMM.

Currently, ASR technology is employed in a diverse array of applications, including but not limited to virtual assistants, speech-to-text transcription, voice search, and language translation. The field of ASR technology remains a dynamic domain of investigation, with persistent endeavors to enhance the precision, resilience, and efficacy of ASR frameworks.

As of 2001, the accuracy rate of speech recognition technology had reached 80%. Throughout the majority of the decade, there were limited technological advancements until the introduction of Google Voice Search in the 2010s. This innovation facilitated speech recognition for a vast number of individuals via an application and delegated processing capabilities to data centers. Google has utilized data from a vast number of searches to enhance the precision of its services. Specifically, its English Voice Search System has integrated a corpus of 230 billion words. During this period, voice recognition applications such as Apple's Siri[1] were introduced, leading to a rise in consumer acceptance of conversing with machines through devices such as Amazon's Alexa[2] and Google Home[3]. Currently, there is a competition among prominent technology corporations to attain the most precise speech recognition system, with Google asserting a minimal error rate of 4.9 percent. The graphical representation presented herein has been extracted from Mary Meeker's 2017 Internet Trends report. The Figure 1 depicts the word accuracy rate of Google, which has recently surpassed the 95% benchmark for human accuracy.

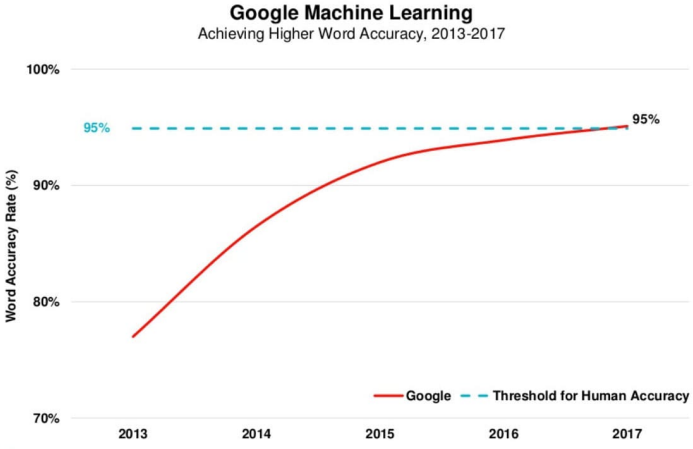

Figure 1: The evolution of Word Accuracy Rate through the years 2013-17

In recent times, the advancement of deep learning-based models, combined with transfer learning methodologies, has contributed significantly to the progress of ASR technology. Deep learning is a machine learning approach that employs intricate neural networks to perform data processing and analysis. Research has demonstrated its remarkable efficacy in the domain of speech recognition, as evidenced by studies conducted by Hinton et al. (2012)[4] and Hannun et al. (2014)[5]. The utilisation of transfer learning in the instruction of ASR models for languages with limited resources not only enhances their efficacy but also surmounts the obstacles of inadequate data by utilising the abundant resources accessible for more prominent languages. Furthermore, transfer

[1] https://www.apple.com/siri/
[2] https://alexa.amazon.com/
[3] https://home.google.com/welcome/

learning facilitates the customization of models to the particular linguistic features and accents of low-resource languages, thereby augmenting the precision and resilience of ASR systems across a range of linguistic settings. The advancement of ASR technology has been facilitated by the increased accessibility of voluminous datasets and the enhanced computational capabilities of contemporary computing systems, commencing from 2014. Convolutional Neural Networks (CNNs) and Recurrent Neural Networkss (RNNs) are currently employed in the field of speech recognition. Specifically, CNNs are utilized for efficient feature extraction from speech signals [6], while RNNs are applied for sequence modeling and classification. The utilization of transfer learning methodologies has been employed to enhance the performance of ASR. This involves training a model on a vast dataset and subsequently fine-tuning it on a smaller dataset that is specific to the task at hand [7]. ASR technology is currently being employed in various industries such as healthcare, legal, automotive, and education. Recent advancements in deep learning and transfer learning have led to substantial progress in ASR technology, resulting in the development of more precise and resilient speech recognition systems. These developments have also created new prospects for ASR technology in diverse applications.

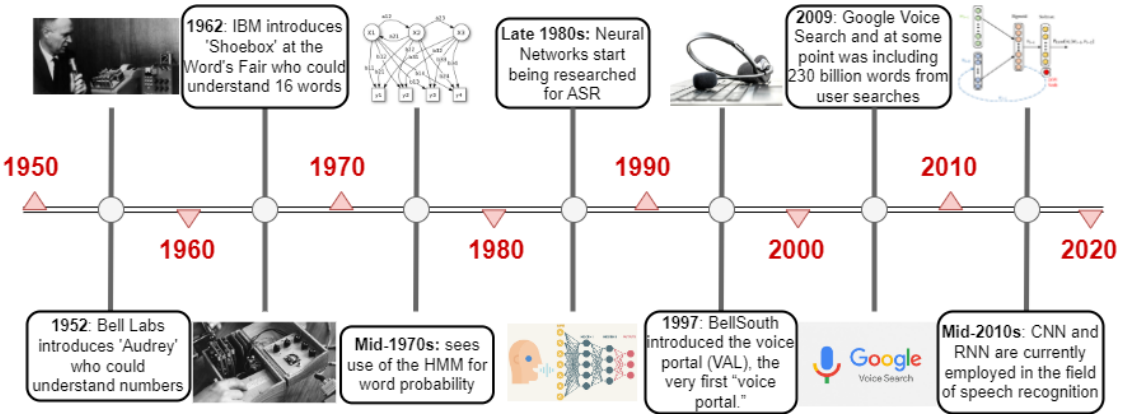

Figure 2: History of Automatic Speech Recognition through the years 1950 - 2020

3.2 Historical Overview of Audio Cleaning

The procedure of audio cleaning, which is intended to enhance the quality of speech signals for the purpose of ASR, has undergone notable progressions throughout the years, as shown in Figure 3. This segment presents a comprehensive historical survey of the development of audio cleaning methodologies, emphasising significant landmarks and contributions within the discipline.

During the initial phases of audio cleaning, the primary emphasis was on fundamental techniques for reducing noise. During the 1970s, scholars initiated an investigation into statistical models, such as spectral subtraction, as a means of mitigating noise interference in speech signals[8, 9]. The aforementioned methodology entails the estimation of the noise spectrum and its subsequent subtraction from the spectrum of the speech signal that is contaminated with noise, thereby improving the intelligibility of the speech. Although spectral subtraction exhibited promising results at first, it exhibited constraints in addressing non-stationary and reverberant noise environments.

In the following years, progressions in Digital Signal Processing (DSP) methodologies facilitated the emergence of more intricate techniques for audio purification. During the 1980s, the noise reduction capabilities were enhanced by the emergence of adaptive filtering algorithms, such as the Wiener filter, which were designed to adapt to the statistical properties of the noise and speech signals[10]. The utilisation of these techniques resulted in a more resilient noise reduction and facilitated the enhancement of ASR accuracy.

The emergence of machine learning and artificial intelligence has brought about a significant transformation in the field of audio cleaning techniques. The utilisation of neural networks for the purpose of noise reduction and speech enhancement has been investigated by researchers. During the 1990s, there was an introduction of RNN and TDNN which aimed to capture temporal

dependencies and contextual information in speech signals. This development led to notable improvements in noise reduction and speech quality[11].

The field of audio cleaning has undergone a significant transformation in recent years, owing to the advent of deep learning models. The utilisation of CNN and Long Short-Term Memory (LSTM) networks has been extensively applied in the domain of noise reduction and feature enhancement. The aforementioned models demonstrate exceptional proficiency in acquiring intricate patterns and interdependencies within speech signals, thereby facilitating superior noise reduction and improved ASR efficacy[12, 13, 14].

The utilisation of GAN for the purpose of audio cleaning has garnered significant interest. GAN utilise a hybrid approach that involves both discriminative and generative networks to acquire knowledge of the relationship between noisy and clean speech signals. The utilisation of GAN has exhibited encouraging outcomes in the domain of unsupervised audio denoising. Specifically, GAN have been observed to acquire the ability to generate clear speech signals from noisy inputs[15, 16, 17].

The current research contributes to the field of audio cleaning by addressing previously identified limitations and gaps in the literature. In contrast to prior investigations that have concentrated on particular facets, the present study adopts a holistic perspective on audio cleansing by taking into account the wider context of ASR and its influence on the efficacy of cleansing methodologies. This study endeavours to determine the optimal combinations of audio cleaning techniques by means of thorough experimentation, rigorous evaluations, and a systematic review of existing literature. This study investigates the potential synergies resulting from the use of multiple audio cleaning techniques by analysing their respective advantages and trade-offs. The ultimate goal is to contribute to the advancement of more effective and resilient audio cleaning methods.

Additionally, the present study focuses on the correlation between ASR and audio cleaning, acknowledging the pivotal function of audio quality in attaining precise speech recognition. This study provides significant contributions to the fields of audio cleaning by thoroughly examining the challenges and opportunities involved. The study's implications have far-reaching consequences for a variety of applications that depend on ASR technology, including virtual assistants, speech-to-text transcription, and voice search, as it improves the precision and practicality of these systems. To summarise, the study's comprehensive approach, exploration of combination techniques, and practical insights make a significant contribution to the field of audio cleaning. This provides valuable guidance for both researchers and practitioners and enhances the overall performance of ASR systems.

3.3 Primary factors affecting ASR accuracy

The efficacy of ASR systems is significantly contingent upon the caliber of the audio input. There are multiple variables that can impact the fidelity of auditory signals, ultimately resulting in diminished ASR precision. Various factors can impact speech recognition, such as environmental factors, speaker variability, speech styles, pronunciation errors, accents and dialects, and background noise. This subsection delves into each of the aforementioned factors comprehensively, utilizing relevant literature to offer perspectives on their influence on the accuracy of ASR. The objective of this study is to enhance comprehension of the difficulties that may emerge during audio data collection and to offer techniques for purifying and prepping audio data prior to its utilization in training ASR models. The most frequent issues with audio recording are listed below, along with how they may affect ASR functionality.

CHA-1: Background noise The term "background noise" pertains to any undesired auditory stimuli that co-occur with the intended speech signal, and it is a formidable obstacle that poses a significant threat to the precision of ASR systems[18]. The existence of ambient noise can considerably diminish the Signal-to-Noise Ratio (SNR) of the speech signal, resulting in inaccuracies in speech recognition[19]. The phenomenon of speech signal interference by noise can manifest even at low noise levels, owing to the possibility of the noise being present within the frequency range of the speech signal, as noted by Hirsch and Pearce (2011)[20].

The presence of background noise introduces extraneous acoustic energy to the primary

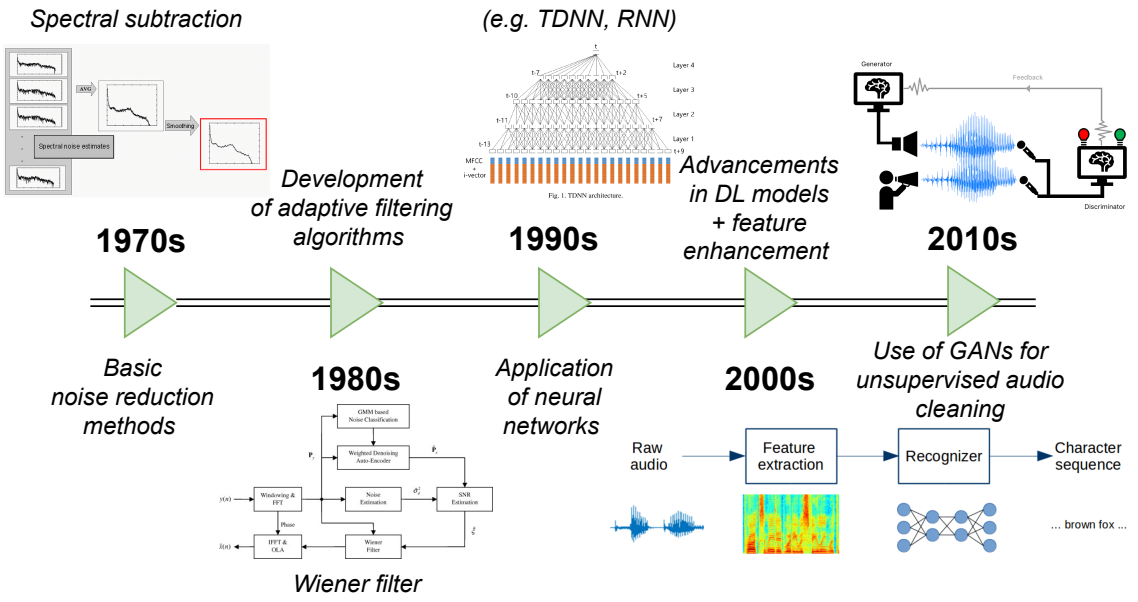

Figure 3: History of Audio Cleaning through the years 1970 - today

speech signal, leading to a modification of the speech waveform. Figure 4 provides an illustration of background noise, wherein the left side of the image exhibits auditory activity while the right side remains devoid of sound. The presence of noise may result in the superimposition of the noise signal onto the speech signal, leading to the possibility of distortion or partial masking of certain segments of the speech signal.

The presence of noise has a differential impact on the amplitude and frequency components of the speech signal, thereby posing a greater challenge for the ASR system to effectively extract the requisite information from the speech signal. Furthermore, the presence of noise has the potential to induce variations in both the amplitude and duration of the speech signal, thereby resulting in alterations in the temporal and spectral attributes of the signal as reported by García-Perera et al. (2020)[18].

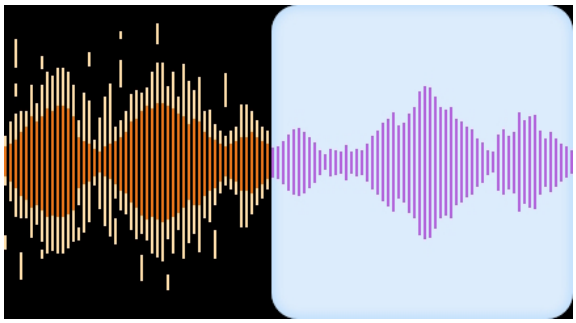

Figure 4: The presence of background noise in an audio file; The left side of the image is audible while the right side is silent.

CHA-2: Speaker variability The term "speaker variability" pertains to the dissimilarities in speech patterns, accent, pronunciation, and other attributes that may differ across various speakers. The existence of speaker variability can present a substantial obstacle for ASR systems, as it may lead to diminished precision and heightened rates of errors. Park and Kim (2019) conducted a study which revealed that speaker variability, particularly in the

case of non-native accents, had a significant impact on ASR performance. Hori et al. (2018) conducted a study to investigate the impact of speaker variability on the accuracy of ASR in a multilingual context. The findings of the study revealed that an increase in the number of non-native speakers resulted in a decrease in the accuracy of ASR systems.

Variations in speech tempo, pitch, and intonation can also arise due to speaker variability. The presence of these discrepancies can result in waveform distortion, thereby affecting the precision of speech recognition mechanisms. An instance of a speaker with a high-pitched voice could potentially generate a waveform that exhibits a dissimilar frequency distribution in comparison to a speaker with a lower-pitched voice. Likewise, a speaker who articulates at a rapid pace may generate a waveform with a dissimilar temporal dispersion in comparison to a speaker who enunciates at a leisurely pace. The fluctuations in frequency and temporal dispersion pose a challenge for speech recognition systems to precisely detect and transcribe speech.
Figure 5 depicts an example of three audio wave files containing the same word spoken by three distinct individuals. Evidently, the three representations exhibit a similar pattern in their respective low and high points. However, there exist notable dissimilarities in certain particulars, such as the magnitude of the pause and the requisite duration for each syllable. These differences are created due to unique pronunciation characteristics that each person has.

CHA-2A: Accents and Dialects The acoustic properties of speech are significantly affected by accents and dialects, which presents a notable obstacle for ASR systems. The presence of accents and dialects can lead to significant variations in speech patterns, including alterations in the duration and stress of phonemes, thereby contributing to the occurrence of ASR errors. Research has indicated that ASR systems' recognition accuracy may decline by as much as 20% when processing non-native accents in contrast to native accents[21, 22]. Scholars have conducted studies on the influence of dialectal variation on the performance of ASR, including the distinctions between British English and American English, as reported by Nina Markl(2022)[23]. The aforementioned studies underscore the necessity of training ASR systems on a varied spectrum of accents and dialects in order to enhance their resilience.

This variation is included for the sake of the paper because the audio files contain words that were recorded in two different dialects. There are several pronunciation differences between the Greek and Cypriot Greek dialects of the Greek language. The way some sounds and letters are pronounced is one notable difference. For instance, the letter "τ" (tau) is frequently pronounced as a "tch" sound in Cypriot Greek, but a "t" sound in Standard Greek. Furthermore, vowel sounds not found in Standard Greek are frequently used in the Cypriot dialect, such as the "u" sound in words like "uranos," which means "sky."

The two dialects' intonation and stress patterns also differ from one another. The intonation of Cypriot Greek frequently has a more sing-song quality and a higher pitch at the ends of sentences. While Standard Greek typically stresses the penultimate (second-to-last) syllable, Cypriot Greek frequently emphasises the final syllable of a word [24].

CHA-2B: Speech Styles ASR systems encounter difficulties in recognising speech patterns that are introduced by various speech styles, such as whispering, shouting, and fast speech, due to their inherent variability. As an illustration, the act of whispering and shouting can have a notable impact on the amplitude and periodicity of speech signals, thereby rendering them more arduous to perceive with precision. Rapid speech, conversely, may cause phonetic information loss and speech sound blending, thereby resulting in errors in recognition. According to Grozdić et al. (2017), research has indicated that the precision of ASR systems may diminish by as much as 25% when attempting to recognise speech that is either whispered or shouted in comparison to speech that is spoken at a normal volume [25]. The field of speech properties encompasses a comprehensive spectrum of vocal modes, including but not limited to whispering, soft speech, normal speech, loud speech, and shouting. The study of Zelinka et al. (2012)

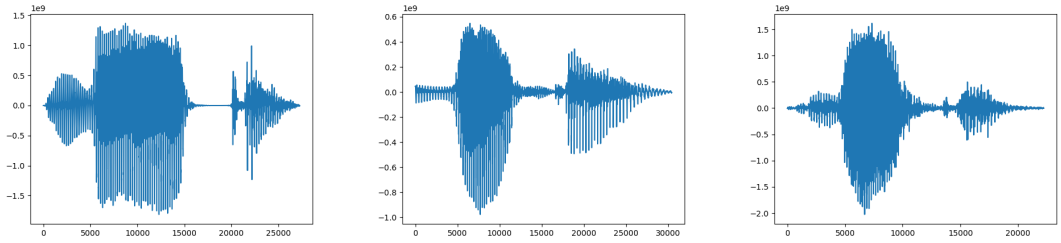

Figure 5: A sample of audio wave files containing three individuals saying the same word.

demonstrates the influence of variability in vocal effort on the efficacy of an isolated-word recognition system[26]. Furthermore, the study evaluates various techniques to enhance the system's resilience. When trained on normal speech, the accuracy for normal speech was 80%. The results of the test indicate a significant reduction in performance, with a decrease of up to 40%, when evaluating whispered speech. This observation provides a rationale for the divergences that may arise among distinct trained models featuring diverse speech patterns.

CHA-3: Environmental factors The performance of ASR systems can also be influenced by environmental factors. The clarity of a recorded speech signal can be impacted by factors such as the quality of the microphone and the acoustics of the room. This can pose a challenge for ASR systems in accurately recognising spoken words. Furthermore, the spatial separation between the speaker and the microphone can potentially deteriorate the ratio of the signal to noise, thereby impeding the efficacy of the ASR system. The degradation of speech signal and the consequent negative impact on ASR performance can be attributed to reverberation, which is the result of sound waves reflecting off various surfaces in the environment.

An instance of the influence of environmental factors on ASR can be observed in the research carried out by Allen et al. (1979)[27]. The study examined the impact of room acoustics on the efficacy of ASR systems. The study conducted by the authors revealed that the precision of ASR systems was considerably affected by the duration of reverberation and the intensity of ambient noise within a given space. The outcomes indicated that heightened levels of reverberation and noise were associated with a decline in the performance of the ASR systems.

Matthias et al. (2009) conducted a study that investigated the impact of microphone type and placement on the accuracy of ASR[28]. The study conducted by the authors revealed that the utilisation of directional microphones in close proximity to the speaker's mouth resulted in a significant enhancement in the performance of ASR systems, in comparison to the utilisation of omnidirectional microphones positioned at a greater distance from the speaker. In order to ensure optimal sound capture in an ASR system, it is imperative that the microphones be situated in a stationary position in close proximity to the sound source, which is typically the speaker's mouth. Accordingly, body-mounted microphones, including headsets and lapel microphones, offer superior sound quality.

Environmental factors can have adverse impacts on the recording, which can be observed through various manifestations in the waveform display. In instances where there exists a notable presence of reverberation, the waveform may exhibit extended decay durations, leading to a potential reduction in the lucidity and comprehensibility of the speech signal. Figure 6 depicts an illustration of this phenomenon. In the event that the recording is captured remotely from the speaker, the waveform may exhibit a decreased amplitude and an increased level of noise. This can pose a challenge in discerning the speech signal from ambient noise.

CHA-4: Pronunciation errors The transcription of speech into text in ASR systems is contingent upon the precision of the acoustic models utilised. However, when a speaker pronounces

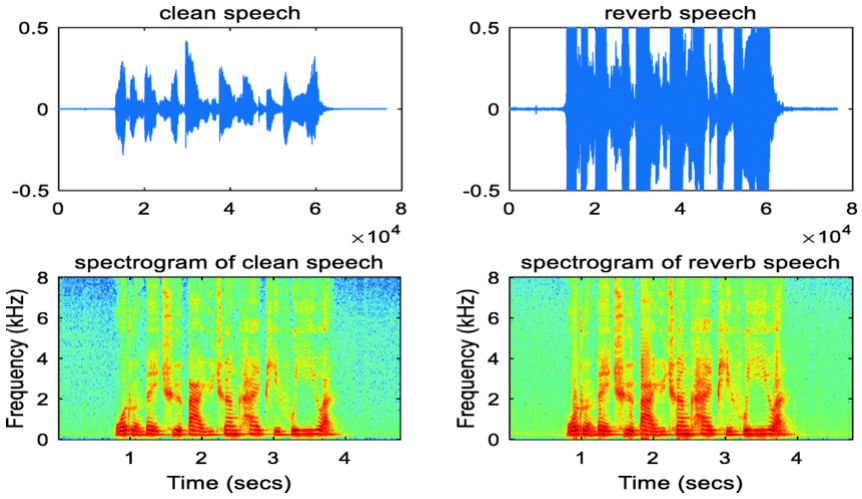

Figure 6: Effect of reverberation on speech waveform and spectrogram from El-Moneim's et al.(2020) paper[29]

a word differently from the way it is pronounced in the training data, ASR accuracy can be significantly affected. This is known as a pronunciation error. There are multiple variables that can potentially influence the occurrence of pronunciation inaccuracies, such as the speaker's mother tongue, accent, and manner of speaking. Studies have shown that non-native speakers of a language are more likely to make pronunciation errors, especially when they are less proficient in the language (Gibbon et al.)[30]. Numerous research studies have explored the influence of pronunciation inaccuracies on the precision of ASR systems. For example, one study by Lee and Hon (2019) found that mispronunciation of specific phonemes resulted in a significant decrease in ASR accuracy, particularly for non-native speakers[31]. Pronunciation errors can affect the waveform in several ways. Initially, the waveform may exhibit a deficiency in distinctness or accuracy in the articulation of specific phonemes or lexemes, resulting in a more erratic configuration. This can result in the waveform being harder to distinguish, and thus harder for ASR systems to accurately interpret. Moreover, the presence of mispronounced words or sounds can result in the manifestation of entirely distinct words or sounds in the waveform, thereby causing perplexity in ASR systems. The aforementioned phenomenon may be interpreted as a modification in either the frequency or amplitude of specific segments of the waveform, which can result in imprecisions in the process of speech recognition. The occurrence of pronunciation inaccuracies can potentially generate additional interference in the waveform, thereby impeding the ASR system's ability to differentiate between the intended speech and the erroneous phonetic or lexical units. This can result in a more complex waveform with more variability in amplitude and frequency, further complicating the speech recognition process.

4 Methodology

The methodology utilized in acquiring the 18 recordings utilized in this research involved a focused approach to crowdsourcing. Ideally, a greater quantity of recordings would have been preferable in order to encompass a wider spectrum of speech variations and enhance the statistical potency of the analysis. Notwithstanding practical and temporal limitations, as well as the extent of this preliminary stage, the present dataset provides a basis for investigating the utilization of audio cleansing methods in the context of automated speech recognition.

In order to obtain the recordings, participants were instructed to either access the mobile application or navigate to the website, `https://dagbani-speak.web.app/`. All participants were given instructions to record their speech of all the assigned words at a minimum of one instance. The process of recording was conducted in the participants' self-selected environment and at their preferred time, thereby facilitating a recording experience that was both natural and comfortable.

Measures were taken to ensure speaker diversity through the recruitment of individuals from a variety of backgrounds. The sample size comprised 18 individuals, with an equal distribution of 10 females and 8 males, thereby ensuring gender balance in the study as shown in left side of Figure 7. Furthermore, the participants consisted of individuals from different age groups, allowing for variations in speech patterns and characteristics. Among the participants, there were 2 Greek speakers, 2 international speakers proficient in the language, and 14 Cypriot speakers, as Table 1 shown. This combination of speakers with different linguistic backgrounds contributes to the richness and diversity of the collected recordings.

Type	Number of people
Greek Speakers	2
Cypriot Speakers	14
International Speakers	2

Table 1: Population of people with different dialects and accents.

By employing targeted crowdsourcing and incorporating a variety of speakers, this methodology aimed to capture a comprehensive dataset that accurately represented the speech characteristics relevant to the specific context of the study. The inclusion of participants from different age groups and linguistic backgrounds enhances the validity and generalizability of the findings, providing valuable insights into the effectiveness of the audio cleaning techniques for automatic speech recognition in a diverse range of scenarios.

Targeted crowdsourcing, while a valuable approach for data collection, does come with its own set of challenges. One of the primary challenges is to ensure a sufficient number of participants who fulfill the particular criteria of the research. The process of identifying individuals with the requisite language proficiency, accent, or other desired attributes can be a laborious task that necessitates meticulous recruitment tactics. Additionally, the level of engagement and commitment from participants can vary, as they are contributing their recordings in their own time and environment. The decentralized approach to data collection may give rise to inconsistencies in recording standards, ambient interference, or other contextual factors that could potentially affect the integrity of the entire dataset. Notwithstanding the difficulties, focused crowdsourcing persists as a viable and efficient approach for collecting a varied assortment of recordings that mirror genuine speech situations, thereby allowing researchers to scrutinize and assess audio refinement techniques for automated speech recognition in a more all-encompassing fashion.

Moreover, to guarantee a fair and impartial portrayal and mitigate partialities, significant emphasis was placed on procuring a nearly equivalent quantity of recordings from every participant, as shown in right side of Figure 7. The objective of this approach was to accommodate possible discrepancies in speech patterns, pronunciation, and vocal traits across diverse individuals. The dataset was augmented with varied samples by gathering an adequate number of recordings from each participant, enabling a thorough assessment of audio cleansing methodologies across multiple speakers. The equitable allocation of recordings augments the dependability and soundness of the study's conclusions, furnishing a sturdy foundation for evaluating the efficacy of various methodologies in improving the performance of automated speech recognition.

5 Audio Enhancement Techniques for ASR Systems

Subsection 3.3 elucidates that there exist several factors, namely **CHA-1** to **CHA-4**, which can exert a significant impact on the efficacy of ASR systems. In order to address these challenges, scholars have devised diverse methodologies to alleviate the influence of these factors on ASR efficacy. The present chapter aims to examine various techniques and evaluate their efficacy in enhancing ASR accuracy in challenging acoustic conditions. The two broad categories of these methods are feature-based and model-based approaches. Feature-based methodologies encompass the extraction of more resilient features from the speech signal, whereas model-based methodologies involve the training of more advanced models that can more effectively manage the variability in

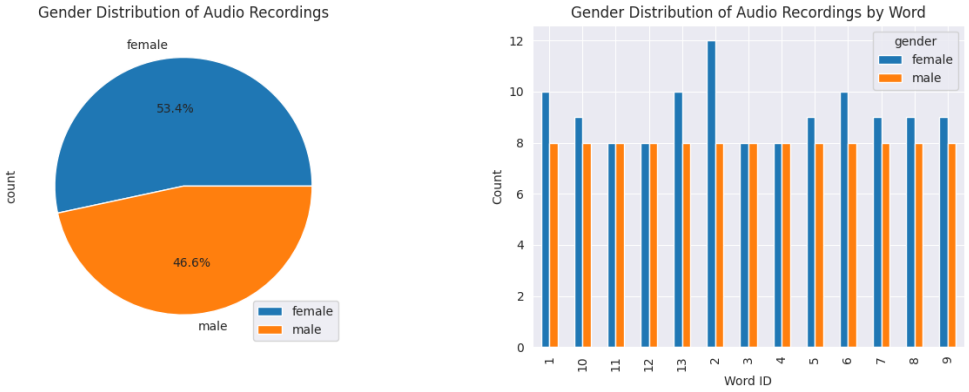

Figure 7: Left: Pie chart illustrating the distribution of genders among the recorded participants. Right: Column chart depicting the number of word recordings contributed by each gender.

the speech signal. The subsequent section will delve into the intricacies of these methodologies and analyse their respective merits and drawbacks.

5.1 Exploring Essential Libraries

This section will delve into the code implementation, with a particular emphasis on the libraries employed. Libraries are essential in addressing diverse tasks, including but not limited to data analysis, audio processing, visualisation, and other related functions. Libraries offer a range of functionalities and tools that streamline the development process and optimise the overall efficiency of the code. This paper will provide a comprehensive analysis of the libraries in question, encompassing their intended function, characteristics, and their role in fulfilling the distinct demands of the project.

1. **soundfile (Source: PySoundFile, Developer: Bastian Bechtold)**: This library provides an interface to read and write audio files. In the code, soundfile is imported as sf, and it is used to read the audio files for further processing.

2. **re (Regular Expressions, Source: Python Standard Library, Developer: Python Software Foundation)**: The re module allows you to work with regular expressions in Python. In the code, it is used to extract ID numbers from the file paths and file names.

3. **pandas (Source: pandas, Developer: pandas Development Team)**: Pandas is a powerful library for data manipulation and analysis. In the code, it is used to create and manipulate DataFrames. It is utilized it to perform operations such as grouping, counting, and visualizing data.

4. **seaborn (Source: seaborn, Developer: Michael Waskom)**: Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics. In the code, seaborn is used to create count plots and bar charts to visualize the frequency and distribution of categories, words, and gender in the audio recordings.

5. **matplotlib (Source: Matplotlib, Developer: Matplotlib Development Team)**: Matplotlib is a widely used plotting library in Python. It provides a flexible and comprehensive set of functions for creating static, animated, and interactive visualizations. In the code, matplotlib is used to create various plots, such as bar charts, count plots, and pie charts, to visualize the distribution of categories, words, and gender in the audio recordings.

6. **firebase_admin (Source: Firebase Admin SDK, Developer: Firebase team)**: Firebase Admin SDK allows you to interact with Firebase services from a server environment.

In the code, firebase_admin is used to initialize the Firebase Admin SDK and interact with the Firebase storage service. It enables the downloading of files from a specified bucket and folder path.

7. **google.cloud.storage (Source: google-cloud-storage, Developer: Google)**: The google.cloud.storage library provides a client interface for interacting with Google Cloud Storage. In the code, it is used to work with the storage bucket and list the files in a specified folder path.

8. **os (Source: Python Standard Library, Developer: Python Software Foundation)**: The os module provides a way to interact with the operating system. In the code, it is used to create directories for storing downloaded and modified files.

9. **pickle (Source: Python Standard Library,Developer: Python Software Foundation)**: The pickle module allows you to serialize Python objects to a byte stream and deserialize them back into objects. In the code, it is used to store and load the metadata list as a pickle file.

10. **pydub (Source: pydub, Developer: James Robert)**: PyDub is a simple and easy-to-use library for audio processing in Python. In the code, pydub is used to load, manipulate, and export audio files. It provides functions for applying noise reduction, compression, silence detection, and normalization to the audio data.

11. **numpy (Source: NumPy, Developer: NumPy developers)**: NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. In the code, numpy is used for various operations, such as converting audio data to NumPy arrays, computing score values, and performing mathematical computations.

12. **wave (Source: Python Standard Library, Developer: Python Software Foundation)**: The wave module provides a convenient interface to the Waveform Audio File Format (WAV) file format. In the code, it is used to open and read the audio files in WAV format for further processing.

5.2 Mitigating the factor CHA-1

The presence of ambient noise poses a significant obstacle for ASR systems, as it can considerably diminish the precision of speech recognition. There are various techniques that can be employed to mitigate ambient noise in audio recordings intended for ASR purposes. Utilising noise reduction algorithms is a viable approach for mitigating background noise in audio files intended for ASR purposes. The algorithms function by conducting an analysis of the audio file and discerning the ambient noise, subsequently eliminating or decreasing it while maintaining the integrity of the speech signal. Numerous noise reduction algorithms exist, such as spectral subtraction, Wiener filtering, and adaptive filtering. Utilising a DSP technique that entails the application of a low-pass filter and a high-pass filter to individual segments of audio is a highly efficacious approach for eliminating background noise from speech signals. This methodology can prove to be especially advantageous in instances where speech signals are marred by broadband noise or persistent noise throughout the recording. Hence, it is deemed as the optimal choice given that one of the aims of this project is to enhance the quality of speech signals for the purpose of advancing ASR.

The purpose of the low-pass filter is to eliminate high-frequency noise that could potentially exist in the signal, such as electrical noise or hiss. The high-pass filter is designed to remove low-frequency noise that may be present in the signal, such as hum or rumble. The utilisation of dual filters enables the retention of a spectrum of frequencies located in the central region, which is known to contain significant speech-related data, while simultaneously eliminating noise present at the extremities of the spectrum.

5.2.1 Implementation mitigating techniques for CHA-1

Listing 1 presents the noise reduction Python code about a function called "noise_reduction" that performs noise reduction on an audio file and saves the filtered audio as a new WAV file. The code first extracts the audio data from the input audio file as a numpy array, while also retrieving information about the file, such as sample rate, number of channels, and sample width.

Subsequently, the audio data is reshaped into a two-dimensional array, where each row represents a channel. Using the "split_on_silence" function from the "pydub" library, the code splits the audio file into chunks based on detected periods of silence. By splitting the recording into smaller chunks and applying the filters to each chunk separately, the noise reduction can be more targeted and effective.

Next, for each chunk of audio, the code applies a low-pass filter to remove high-frequency noise above 4000 Hz and a high-pass filter to remove low-frequency noise below 200 Hz, using the "low_pass_filter" and "high_pass_filter" functions from the "pydub" library. The filtered chunks are then concatenated back into a single audio file.

Finally, the filtered audio is saved as a new WAV file in a directory named "modified_files", with the name of the new file based on the input file name, with "_noise_red" appended to the end.

```
def noise_reduction(audio,name):

    # Extract the audio data as a numpy array
    audio_data = np.array(audio.get_array_of_samples())
    sample_rate = audio.frame_rate
    num_channels = audio.channels
    sample_width = audio.sample_width
    audio_data = np.array(audio.get_array_of_samples())

    # Convert the audio data to a numpy array
    audio_data = np.frombuffer(audio_data, dtype=np.int16)

    # Reshape the audio data into a two-dimensional array (one row per channel)
    audio_data = audio_data.reshape(-1, num_channels)

    # Split the audio file into segments using silence detection
    chunks = split_on_silence(audio, min_silence_len=10, silence_thresh=-30)

    # Apply noise reduction to each chunk using the built-in filter function
    for i, chunk in enumerate(chunks):
        filtered_chunk = chunk.low_pass_filter(4000)
        filtered_chunk = filtered_chunk.high_pass_filter(200)
        chunks[i] = filtered_chunk

    # Concatenate the filtered chunks back into a single audio file
    filtered_audio = None
    if chunks:
        filtered_audio = chunks[0]
        for chunk in chunks[1:]:
            filtered_audio = filtered_audio + chunk

    if filtered_audio is not None:
        # Export the filtered audio to a new WAV file
        filtered_audio.export("./modified_files/" + name + "_noise_red.wav", format
    ="wav")
    else:
        audio.export("./modified_files/" + name + "_noice_red.wav", format="wav")
```

Listing 1: Python code: Noise reduction function

Listing 2 contains the code for a function that applies an oversmoothing filter to an audio signal using a moving average filter. This function applies oversmoothing using a moving average filter to an input audio signal specified as an ndarray in the audio_signal argument. The window_size parameter determines the degree of oversmoothing by specifying the size of the moving average window. window_size is set to 5 by default and is defined as a numpy array of ones divided by the window size. This results in a uniform weighting of samples within the window. The scipy.signal library's lfilter function is used to apply the moving average filter to the input audio signal. As

filter coefficients, the window array is utilised, and a normalisation factor of 1 is specified for the denominator. The output is stored in the variable oversmoothed_signal. The final step is to export the oversmoothed audio signal as a WAV file using the export() method. The file is saved in the./modified_files/ directory with the name "_overmoothing.wav" appended to the original name argument.

```
def oversmoothing(audio,name, window_size=5):
    audio = np.asarray(audio)
    audio = audio.squeeze()  # Remove single-dimensional entries from the shape of
    the array

    window = np.ones(window_size) / window_size
    oversmoothed_signal = lfilter(window, 1, audio)
    oversmoothed_signal.export("./modified_files/" + name + "_overmoothing.wav",
    format="wav")
```

Listing 2: Python code: Oversmoothing function

5.2.2 Fine-Tuning Low-Pass and High-Pass Filters

The present code employs a low-pass filter for the purpose of eliminating high-frequency noise that surpasses the threshold of 4000 Hz. This approach has been found to be efficacious in mitigating auditory disturbances emanating from sources such as electronic noise or ambient conversations. The high-pass filter is utilised for the purpose of eliminating low-frequency noise that falls below the threshold of 200 Hz. This technique has been found to be efficacious in mitigating noise emanating from various sources, including but not limited to wind, traffic, and air conditioning.The selection of cutoff frequencies for low-pass and high-pass filters is aimed at attenuating noise to a maximum extent while retaining the intended audio signal. The selection of the cut-off frequencies for low-pass and high-pass filters, namely 4000 Hz and 200 Hz, respectively, is determined through an examination of the frequency spectrum of the audio file. This process involves identifying the frequency range in which the noise exhibits the highest prominence. The frequency spectrum of the audio file was visualised through the utilisation of MATLAB[4] software. Achieving a balance between minimising noise and maintaining the integrity of the audio signal is a crucial consideration. Consequently, a prevalent approach employed to determine the optimal equilibrium for audio files involves experimenting with numerous pairs. The filters were subjected to a variety of spectrums, and the most favourable points were selected. The low-pass range comprises the values [2000, 3000, 3500, 4000, 4500, 5000, 6000], whereas the high-pass range encompasses the values [50, 100, 150, 200, 250, 300, 350, 400, 500].

5.2.3 Observations after mitigating the factor CHA-1

Prior to the implementation of noise reduction techniques, the waveform representation of the audio file displayed numerous anomalies that posed challenges in identifying the speech signal. The graph exhibited noteworthy fluctuations in magnitude and a considerable amount of ambient interference that obscured the vocal signal. The discernment of speech segments from ambient sound was found to be arduous due to their apparent submergence within the noise. Furthermore, the noise exhibited a broad spectrum, encompassing both high and low frequencies, thereby exacerbating the discernment of the speech signal. The red lines in the visual representation indicate the uppermost and lowermost points of the ambient noise, particularly at the onset and conclusion of the recording. Meanwhile, the green region denotes the segment of the recording that exclusively comprises background noise beyond the absence of sound.

Upon the implementation of noise reduction techniques, the waveform representation of the audio file exhibited substantial enhancements in both clarity and signal-to-noise ratio. The signal's amplitude exhibited a greater degree of uniformity, with reduced fluctuations and an overall smoother profile. The ambient noise was considerably diminished, thereby enhancing the prominence of the speech signal. The speech segments have been observed to manifest as identifiable and distinct patterns on the graph, exhibiting clearly defined peaks and valleys that correspond

[4] https://www.mathworks.com/products/matlab.html

to the vocalisations of the speaker. Additionally, the audio signal's frequency spectrum exhibited a greater concentration around the speech frequencies, as evidenced by the absence of certain portions at the beginning and end of the signal, with fewer noise components present in the high and low frequency ranges.

In general, the process of noise reduction had a notable effect on the waveform representation of the audio recording, rendering the speech signal more readily distinguishable and amenable to analysis. The findings indicate that the noise reduction methods employed were successful in eliminating a substantial portion of the ambient noise while retaining the fundamental characteristics of the speech signal, as evidenced by the enhanced clarity and signal-to-noise ratio. The wave graph that ensues from the aforementioned process serves as a valuable visual aid in evaluating the audio file's quality and gauging the efficacy of the noise reduction technique.

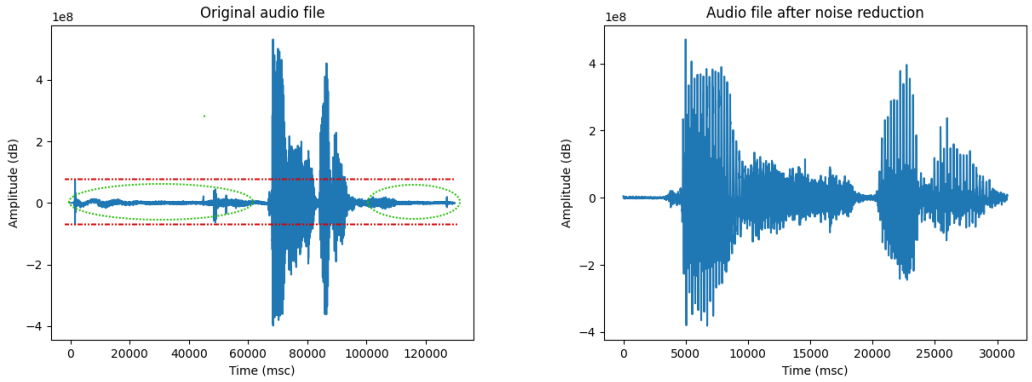

Figure 8: Changes in the audio wave spectrum after noise reduction has been applied [red: maximum point of noise outliers; green: area containing noise outliers].

5.3 Mitigating the factor CHA-2

Speaker variability is a prevalent challenge encountered when working with ASR systems. There are several methodologies that can aid in addressing this issue. **Signal normalisation** is a technique that can be employed to mitigate speaker variability. The term "signal normalisation" pertains to the procedure of altering an audio signal in a manner that results in a more uniform amplitude and frequency distribution. The utilisation of this technique can prove to be advantageous for ASR systems, as it has the potential to mitigate the influence of speaker variability on the precision of the system. The normalisation of the signal enhances the system's ability to identify the phonemes and words uttered, irrespective of the speaker's accent, dialect, or speech style.

There are various methodologies for signal normalisation, each possessing unique merits and demerits. Cepstral Mean Normalisation (CMN) is a frequently employed method in signal processing [5]. It entails the subtraction of the average value of the cepstral coefficients of a signal from every frame of the signal. This technique has the potential to mitigate the influence of fluctuations in the speaker's vocal tract length and other variables that may impact the frequency distribution of the signal. Feature Space Maximum Likelihood Linear Regression (fMLLR) is a technique that entails the training of a regression model to map the acoustic features of a signal to a standard reference signal[6]. This technique can aid in accounting for variations in the speaker's articulation and other variables that may impact the phonetic characteristics of the signal.

The technique of CMN is a commonly employed and straightforward method that entails the deduction of the mean of the feature values over time for every frequency band. The implementation of this methodology has the potential to mitigate channel inconsistency and enhance resilience across diverse recording scenarios. The computational efficiency of CMN renders it a versatile tool that can be seamlessly integrated into any speech recognition system without necessitating supplementary training.

[5] More information about CMN `https://speechpy.readthedocs.io/en/latest/content/postprocessing.html`
[6] More information about FMLLR: `https://dbpedia.org/page/FMLLR`

In contrast, fMLLR is a sophisticated approach that entails the training of a transformation matrix tailored to the speaker, which serves to map the feature space onto a more discerning space. The application of fMLLR has the potential to facilitate the adaptation of an extant acoustic model to a particular speaker or environment, thereby leading to a discernible enhancement in recognition precision. Nevertheless, the utilisation of fMLLR necessitates training data that is specific to the speaker and may incur significant computational costs.

The selection of a suitable normalisation technique is contingent upon the particular demands of the application and the resources at hand. The present project has opted for the utilisation of the CMN technique, which is considered a suitable initial approach for speech recognition systems due to its ease of implementation and potential to enhance recognition accuracy across various scenarios.

Utilising a **varied range of speech data** presents a potential benefit in enhancing the overall precision of the system. This phenomenon can be attributed to the fact that the system's proficiency in speech recognition improves with an increase in the amount of speech data it is trained on. Through the exposure of the system to a diverse array of vocal characteristics and speech modalities, it enhances its capacity to acclimatise to novel voices and speech patterns that it may encounter within the actual context.

Moreover, incorporating a varied range of speech data can enhance the end-user's experience of the ASR system. The reason for this is that users exhibit a higher likelihood of contentment with a system that demonstrates precise speech recognition capabilities, irrespective of their accent or dialect. Through the process of training the system on a varied and inclusive corpus of speech data, the system's ability to accurately identify and comprehend a broader spectrum of speech patterns and styles can be enhanced. This, in turn, can lead to an improved user experience and increased accessibility for a more diverse user base. The reason for this is that the characteristics of speakers can exhibit substantial variation from one individual to another. Consequently, if an ASR system is trained on a restricted corpus of speech data, its performance may be suboptimal when processing speech from different speakers. The incorporation of varied speech data into the training set can enable ASR developers to ensure that the system is capable of recognising a broad spectrum of speech variations and accommodating the diverse speaking styles of users. Furthermore, the incorporation of a varied training set can enhance the overall generalizability and resilience of the ASR system, thereby increasing its efficacy in practical scenarios and expanding its user base. To succinctly encapsulate, the inclusion of a varied training set is of paramount importance in the development of an ASR system that can effectively discern speech from a diverse pool of speakers and adjust to the intricacies inherent in human language.

5.3.1 Implementation mitigating techniques for CHA-2

Listing 2 presents the CMN algorithm in Python code that performs normalization on an audio file and saves the filtered audio as a new WAV file. The objective of this code is to mitigate the impact of speaker variability by implementing CMN on an audio signal. The initial step of the process involves the conversion of the input audio file into a numpy array, which is achieved through the utilisation of the get_array_of_samples() function from the pydub library. Subsequently, the frame rate of the initial audio signal is obtained by utilising the frame_rate attribute.

Subsequently, the mean of the complete signal can be calculated by utilising the np.mean() function from the numpy library. Subtraction of the mean value from each frame of the audio signal is performed by subtracting the mean of the numpy array from the original signal utilising the - operator.

The CMN algorithm is capable of enhancing the performance of ASR systems by eliminating channel-specific fluctuations in the signal through the subtraction of the signal's mean. The resultant signal is a standardised rendition of the initial signal, which is anticipated to be less impacted by variations in the speaker's characteristics.

Ultimately, the altered signal is reconstituted as a pydub AudioSegment by means of the to-bytes() function from the numpy array, and subsequently persisted as a novel audio file via the export() method.

The utilisation of the pydub library in Python for implementing CMN presents a straightforward and effective approach for the normalisation of an audio signal. The utilisation of numpy arrays

and the pydub AudioSegment class is substantiated by their efficacy as data structures for the manipulation of audio data in the Python programming language. The utilisation of the tobytes() method for the purpose of converting the altered numpy array into a compatible format for the pydub library is a straightforward and efficient approach.

```
def normalization(audio,name):
    # Convert the audio to a numpy array
    signal = np.array(audio.get_array_of_samples())
    rate = audio.frame_rate

    # Apply CMN across the entire audio signal
    signal_cmn = signal - np.mean(signal)

    # Convert the modified signal back to a pydub audio segment
    modified_audio = AudioSegment(signal_cmn.tobytes(), frame_rate=rate,
    sample_width=2, channels=1)

    modified_audio.export("./modified_files/" + name + "_normalizedCMN.wav", format
    ="wav")
```

Listing 3: Python code: Normalization function

Train ASR systems on diverse speech data
In addition through extensive and diverse dataset training, ASR systems can effectively accommodate the multifarious manners in which individuals communicate, encompassing variations in pronunciation, regional language, and speaking patterns. The utilised recording set for this project encompasses a heterogeneous group of speakers, comprising individuals with varying gender identities, age ranges, and socio-cultural backgrounds. The recording set comprised of both male and female speakers, spanning across various age groups from young children to older adults. Furthermore, the recording set comprised participants from Greece and Cyprus, thereby ensuring a diverse range of dialects and speech patterns.

5.3.2 Observations after mitigating the factor CHA-2

The CMN technique is employed to normalise a signal by adjusting the amplitude values in order to mitigate the variability across the signal as it is displayed in Figure 9.

Prior to CMN normalisation, the decibel level exhibit an increase owing to the broader spectrum of amplitude values. Following the process of CMN normalisation, decibel level to decrease as a result of the diminished amplitude value range. Upon the application of CMN (i.e., cepstral mean normalisation) to an audio signal, it can be observed that the mean component of the signal is effectively eliminated. Consequently, all amplitude values exceeding the mean will be transformed into negative values. The outcome of this phenomenon lead to a modification in the total magnitude of the signal, whereby certain samples exhibit reduced values while others exhibit amplified values. The decrease in decibel level does not necessarily imply a degradation of information or signal quality. Instead, it signifies a decrease in variability that can enhance the accuracy of ASR.

It is noteworthy that the preservation of the relative amplitudes of distinct signal components is crucial during the normalisation process. It is imperative that the form and attributes of the signal are maintained, notwithstanding any alterations in the overall magnitude. The observed outcome can be attributed to the uniform application of modifications by CMN across all signal constituents, irrespective of their initial magnitude.

5.4 Mitigating the factor CHA-3

The precision of ASR systems can be impeded by environmental factors, particularly in a crowdsourcing context where users have the liberty to record audio files at their discretion, without stringent regulation over the recording devices or the environment. The quality of speech signals can be negatively impacted by various acoustic distortions such as background noise and reverberation, which can pose challenges for ASR systems in accurately transcribing them.

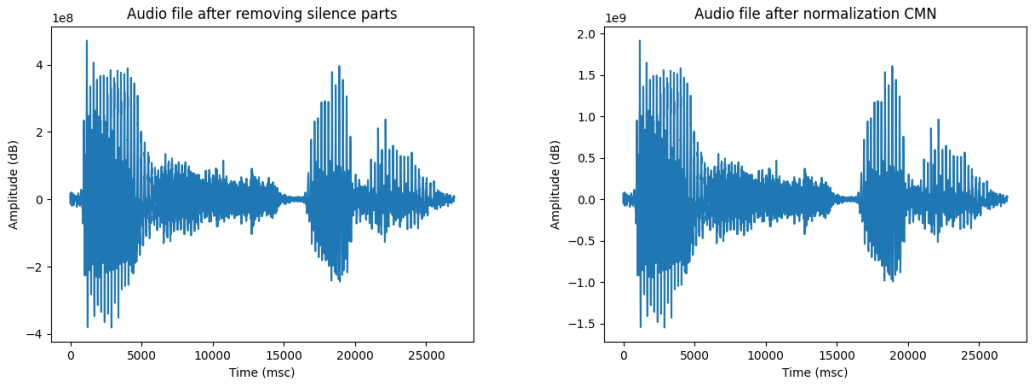

Figure 9: Changes in the audio wave spectrum after limiting the effect of speaker variability.

As a result, the management of environmental variables presents a considerable obstacle. The following justifications examine the inescapable influence of environmental variables on audio data for our ASR system, given the wide array of recording apparatuses and user surroundings.

The impact of uncontrollable recording environments on audio quality cannot be ignored, as factors such as background noise, reverberation, and ambient sounds inevitably affect the final output. Despite the utilisation of noise reduction techniques and sophisticated algorithms, the complete eradication of these factors from audio files remains a challenging task.

1. **Unmanageable Recording Environments:** The utilisation of diverse recording devices by users in our specific crowdsourcing system may result in variations in microphone characteristics and sensitivity levels. The audio quality and susceptibility to environmental factors may vary due to the distinct audio capturing mechanisms employed by these devices. It is imperative for the ASR system to exhibit robustness in order to effectively manage such inconsistencies.

2. **Variation in Recording Equipment:** The practise of enabling users to record audio at their discretion and in any location presents a convenient option, albeit with the drawback of restricted control over the surrounding environment during the recording process. Various factors, such as ambient conversations, vehicular noise, and personal speaking habits such as low volume or distance from the recording device, can considerably influence the quality of audio and, consequently, the accuracy of ASR systems.

3. **User Conduct and Recording Practises:** Acknowledging and addressing the challenges presented by uncontrollable environmental factors in audio files is imperative in our targeted crowdsourcing ASR system. Furthermore, our targeted crowdsourcing system provides users with the option to mitigate and tackle environmental obstacles. Additionally, users are afforded the liberty to review their audio recordings, remove them if deemed necessary, and subsequently re-record them. This functionality enables users to preempt unforeseen occurrences of substandard recordings, affording them an improved prospect of capturing audio of superior quality.

Despite the aforementioned challenges, **the system integrates diverse methodologies** to enhance the influence of ecological elements on the captured audio, as expounded in the preceding subsections 5.2 - 5.4. The aforementioned methodologies encompass normalisation, elimination of background noise, and additional signal processing techniques, as explicated in the preceding subsection. Although these techniques make a significant contribution towards enhancing the audio quality and reducing the impact of environmental factors, it is crucial to acknowledge that they are not a panacea for this problem.

In summary, despite efforts to mitigate the influence of environmental variables, their complete eradication remains a formidable undertaking. Consequently, we are actively pursuing innovations in ASR technology to effectively tackle these obstacles.

5.5 Mitigating the factor CHA-4

The precision of ASR systems, which aim to transcribe spoken language into written text, can be notably affected by inaccuracies in pronunciation. Fortunately, several strategies can be implemented to restrict pronunciation errors in audio files for ASR.

1. **Utilise superior audio recordings**: Utilising high-fidelity audio recordings is imperative for ensuring the precision of a ASR technology. The presence of extraneous noise and distortions in audio recordings of inferior quality can pose a challenge for ASR systems, thereby impeding their ability to effectively identify and transcribe speech. Consequently, it is imperative to utilise audio recordings of superior quality that are devoid of any background noise, distortion, or other forms of interference.

2. **Train ASR systems with a variety of speech samples**: It is customary to train ASR systems on extensive speech datasets to enhance their precision. However, it is advisable to incorporate diverse speech data during the training process. In order to mitigate potential inaccuracies in pronunciation, it is crucial to incorporate a varied assortment of speech data within the dataset utilised for training purposes. The aforementioned may encompass individuals hailing from diverse geographical areas, exhibiting distinct dialects and languages, as well as those with differing speech patterns and impediments.

3. **Use language models**: The utilisation of language models has the potential to enhance the precision of ASR systems through the provision of supplementary contextual information. Language models are trained using extensive datasets of written language and can be utilised to estimate the likelihood of a specific word or phrase occurring within a given context. The integration of language models into ASR systems has the potential to mitigate pronunciation errors by furnishing supplementary context for speech recognition.

To summarise, the mitigation of pronunciation errors in audio files for ASR can be accomplished by utilising high-fidelity audio recordings, varied speech data for training, and the implementation of language models. Through the implementation of these aforementioned techniques, it is plausible to enhance the precision of ASR systems and mitigate enunciation discrepancies.

5.5.1 Implementation mitigating techniques for CHA-4

Prior to exploring the execution of language models, the incorporation of varied speech data, and the choice of audio file formats, it is crucial to underscore the importance of these elements in the creation of resilient and precise ASR systems. Chapter 5.4 outlines three potential methods for addressing environmental factors, which will be elaborated upon below. The project's implementation of these methods will also be discussed.

Selection of audio file format

The selection of the file format can have an impact on the overall quality of the recordings. The Third Generation for mobile Platform (3gp) format may be a viable option for recording audio files within a mobile application, as it presents various advantages.

1. **Compression and file size:** The 3gp file format is frequently utilised for mobile devices due to its compressed nature, which allows for reduced file sizes. The design of this technology is aimed at minimising the size of files without compromising the audio quality to a significant extent. The utilisation of this approach can confer benefits to mobile applications, as it enables the attainment of reduced file sizes, thereby mitigating the storage space demands on the device. Furthermore, it aids in the optimisation of bandwidth consumption during the transmission of audio files across networks, a critical aspect for individuals with restricted data plans.

2. **Device compatibility:** The 3gp format enjoys extensive compatibility with mobile devices, rendering it a dependable option for the development of mobile applications. The utilisation of the 3gp format for audio recording guarantees seamless playback across various mobile devices, irrespective of their hardware specifications or operating systems.

Conversely, in the context of audio recordings for websites, opting for the WAV format may be deemed a viable choice for the ensuing rationales:

1. **Uncompressed audio quality:** The audio file format known as WAV is characterised by its uncompressed nature, which allows it to preserve the entirety of the recorded audio's quality and fidelity. In contrast to compressed file formats such as 3gp, WAV files do not undergo any lossy compression algorithms, thereby yielding a superior level of precision in audio. This holds significant importance for websites that prioritise audio quality, particularly those that feature multimedia content or professional recordings.

2. **Broad compatibility:** WAV files enjoy extensive compatibility as they are widely supported by a majority of web browsers and media players, thus rendering them a dependable option for audio content on websites. The utilisation of the WAV format guarantees seamless playback of audio files across various devices and platforms, devoid of any compatibility discrepancies.

To summarise, the rationale behind opting for 3gp audio files for the mobile application and WAV files for the website is based on a careful analysis of the distinct demands and limitations of each platform. The 3gp format is optimised for mobile devices that have limited storage and bandwidth, providing a compressed format. On the other hand, WAV format offers a high-quality and uncompressed format that is compatible with a broad spectrum of web browsers. Optimising audio quality and compatibility for each platform can be achieved by carefully selecting the appropriate file formats.

Train ASR systems with a variety of speech samples

Chapter 5.3.1 has elaborated on the significance of integrating heterogeneous speech data into ASR systems. In order to guarantee the resilience and versatility of our models, we are currently employing a diverse array of speech data derived from multiple sources and speakers. As a result, this feature aids in mitigating environmental factors.

Employing language models

Incorporating language models into ASR systems offers significant gains in transcription accuracy, contextual understanding, and overall performance. By integrating language models, ASR systems benefit from improved handling of pronunciation errors, enhanced contextual understanding, and better correction of out-of-vocabulary (OOV) words. Language models leverage statistical patterns and contextual information to predict the most probable words or phrases based on the input speech, resulting in more accurate transcriptions. These models also enable ASR systems to capture nuances, idiomatic expressions, and domain-specific terminology more effectively, enhancing language understanding and producing contextually appropriate and linguistically coherent transcriptions.

Furthermore, it is worth mentioning that language models will be employed in a further step of the process when training models are built.

Speech enhancement

Speech enhancement is a technique that can substantially enhance the performance of ASR systems in the project. This technique converts spoken language into written text, and it heavily relies on the quality and intelligibility of the input audio. Speech enhancement techniques have a specific objective of improving the quality and intelligibility of speech signals through the reduction of noise and enhancement of the signal-to-noise ratio. The aforementioned techniques utilize diverse algorithms to attenuate or eliminate undesirable noise while retaining crucial speech data. Speech enhancement has the potential to offer various advantages to ASR systems by mitigating the effects of noise on the audio signal.

1. **Enhanced Accuracy:** Speech enhancement techniques can aid in achieving higher accuracy in recognizing and transcribing spoken words by reducing noise levels and improving the clarity of the speech signal. This is particularly relevant for ASR systems. Improved audio signals enable ASR models to concentrate on speech characteristics and minimize the probability of misapprehension due to noise disruption.

2. **Enhanced Stability:** The implementation of speech enhancement techniques enhances the robustness of ASR systems, particularly in real-world settings that present challenging acoustic conditions. The implementation of noise suppression or elimination techniques facilitates the optimal performance of ASR systems in environments characterized by high levels of ambient noise, such as crowded public spaces, street recordings, or telecommunication applications.

3. **Better Generalization:** Enhanced speech data has been observed to result in improved generalization of ASR models when tested on data that is either noisy or unseen. Through the utilization of speech signals that have undergone noise reduction enhancement during ASR model training, the models acquire greater resistance to noise fluctuations, thereby enhancing their performance when processing unprocessed, real-world audio.

Implementation technique for speech enhancement

Listing 4 contains Python code that implements a speech enhancement technique based on spectral subtraction. Utilizes spectral subtraction to perform speech enhancement on an audio file. The augmented audio is saved with the filename name + "_enhancement.wav" Sample_rate and audio data are extracted from the imported audio file at the outset. The code verifies that the audio has multiple channels (audio.ndim $>$ 1) to assure compatibility with stereo audio. If so, the audio is converted to mono using np.mean to calculate the average of all channels. In order to convert audio data from the time domain to the frequency domain, Short-Time Fourier Transform (STFT) is employed. STFT parameters, including frame size and frame displacement, are specified in seconds. The audio is divided into frames that overlap, and the Fourier transform is calculated for each frame using the fft function from the scipy.fftpack module. The noise spectrum is estimated using the initial 50 frames of the STFT (stft[:50]). This noise estimate is computed by using np.mean to take the mean of the magnitudes of the selected frames. The magnitude of the STFT is subtracted from alpha times the estimated noise spectrum to execute spectral subtraction. The result is then reduced to zero and combined with phase data to reconstruct the improved STFT. Using the Inverse Short-Time Fourier Transform (ISTFT), the STFT is transformed back into the time domain. The enhanced audio is then reconstructed by merging the frames together, taking overlap and padding into account. Using slicing, the reconstructed audio is shortened to the original length of the input audio. The enhanced audio is exported as a WAV file using the export function, with the filename transformed by concatenating "_enhancement.wav" with the name parameter.

```
def speech_enhancement(audio_file, name, alpha=1.0):
    # Load the audio file
    sample_rate, audio = wav.read(audio_file)

    # Convert audio to mono if it's in stereo
    if audio.ndim > 1:
        audio = np.mean(audio, axis=1)

    # Apply Short-Time Fourier Transform (STFT)
    frame_size = 0.025  # Frame size in seconds
    frame_shift = 0.01  # Frame shift in seconds
    frame_length = int(sample_rate * frame_size)
    frame_step = int(sample_rate * frame_shift)
    num_frames = int(np.ceil(len(audio) / frame_step))
    padded_size = num_frames * frame_step
    audio = np.pad(audio, (0, padded_size - len(audio)), 'constant')

    stft = np.empty((num_frames, frame_length), dtype=complex)
    for i in range(num_frames):
        frame = audio[i * frame_step : i * frame_step + frame_length]
        stft[i] = fft(frame)

    # Estimate noise spectrum using minimum statistics
    noise_frames = stft[:50]  # Use the first 50 frames as noise estimate
    noise_spectrum = np.mean(np.abs(noise_frames), axis=0)

    # Apply spectral subtraction
    enhanced_stft = np.maximum(np.abs(stft) - alpha * noise_spectrum, 0) * np.exp(1
    j * np.angle(stft))
```

```
29
30      # Inverse Short-Time Fourier Transform (ISTFT)
31      enhanced_audio = np.zeros(padded_size)
32      for i in range(num_frames):
33          frame = ifft(enhanced_stft[i]).real
34          enhanced_audio[i * frame_step : i * frame_step + frame_length] += frame
35
36      enhanced_audio = np.asarray(enhanced_audio[:len(audio)], dtype=np.int16)
37      enhanced_audio.export("./modified_files/" + name + "_enhancement.wav", format="
        wav")
```

Listing 4: Python code: Speech enhancement function

5.6 Supporting Functions for Audio Enhancement

Apart from the mitigation techniques expounded in sections 5.2, 5.3.1, 5.4, and 5.5, there are a number of auxiliary functions that are instrumental in enhancing the calibre and applicability of audio files for ASR systems. The present subchapter centres on two fundamental operations, namely compression and the removal of silent portions located at the onset and offset of audio recordings. The aforementioned functions play a crucial role in producing audio files that are refined and streamlined, thereby augmenting the precision and efficacy of automated speech recognition procedures.

1. **Compression:** Compression is a technique in digital signal processing that is used to decrease the dynamic range of an audio signal. The mechanism involves the reduction of the amplitude of the high-intensity segments of the audio signal and the amplification of the low-intensity segments, leading to a more homogeneous and equitable auditory experience. The process of compression aids in the standardisation of audio levels, thereby mitigating the occurrence of excessively loud or soft segments in speech. Compression is a technique that reduces the dynamic range of audio signals, thereby mitigating the risk of distortion and clipping. This facilitates the accurate capture and interpretation of speech by the ASR system.

   The significance of compression is rooted in its capacity to augment the comprehensibility and uniformity of speech signals. The process of optimising the audio range facilitates the recognition and transcription of spoken words by the ASR system. In addition, the utilisation of compression techniques aids in mitigating the adverse effects of ambient noise by rendering it more uniform in relation to the audio levels at large. This leads to an enhanced signal-to-noise ratio and consequently, an improved performance of ASR systems.

5.6.1 Implementation technique for compression

The below code in Listing 3 shows the implementation of a function called "compression" that performs dynamic range compression on an audio file. The code starts by defining two parameters, "compression_ratio" and "compression_threshold." The compression ratio determines the amount of gain reduction applied to the audio signal, while the compression threshold specifies the level at which the compression starts to take effect. These values are adjusted based on the specific requirements of the audio file and the desired compression effect. Using those parameters, the code applies dynamic range compression to the input audio. The "compress_dynamic_range" function, likely from an audio processing library, is used to perform the compression. This function adjusts the amplitude of the audio signal, reducing the dynamic range and making softer parts of the audio more audible while limiting the peaks. After applying compression, the resulting audio is saved as a new WAV file. The code exports the compressed audio file to a directory named "modified_files" using the original filename with "_compressed" appended to it. This ensures that the modified audio file is stored separately and can be easily identified.

```
1 def compression(audio,name):
2     # Define the compression ratio and threshold
```

```
    compression_ratio = 4.0
    compression_threshold = -20.0

    # Apply the compression
    output_audio = audio.compress_dynamic_range(threshold=
    compression_threshold, ratio=compression_ratio)

    # Save the output audio file
    output_audio.export("./modified_files/" + name + "_compressed.wav", format
    ="wav")
```

Listing 5: Python code: Compression function

2. **Removal of silent:** The removal of silent or quiet portions from audio files is a common practise as these segments typically include extraneous noise or non-verbal sounds. The presence of silent intervals in speech does not augment the speech content and may result in undesired artefacts or prolonged periods of silence while processing ASR. The elimination of these inactive segments is a pivotal measure in the pre-processing of audio data for ASR systems.

   Through the elimination of the initial and trailing periods of silence, the audio file is efficiently condensed to encompass solely the pertinent speech material. This process effectively removes extraneous interruptions, ambient sounds, or non-verbal utterances, thereby optimising the data for ASR analysis. The act of removing periods of silence serves to not only decrease the computational resources necessary for v, but also to improve the precision and effectiveness of the system by directing attention solely towards the speech segments that contain relevant information. The significance of removing periods of silence is rooted in its ability to offer a more polished and succinct auditory input to the ASR system. The elimination of non-speech segments mitigates the risk of erroneous identification or misapprehension due to extraneous auditory material. Moreover, the act of trimming audio files enhances their overall usability by rendering them more manageable and convenient for storage, transmission, or subsequent analysis.

5.6.2 Implementation technique for removing the silence parts

The code below consists of two functions: "removeSilenceParts" and "detect_leading_silence."

- **removeSilenceParts:** The "removeSilenceParts" function is responsible for removing silence parts at the beginning and end of the audio file. The code first uses the "detect_leading_silence" function, which is explained below, to determine the duration of the silence at the beginning and end of the audio file. Using the information obtained from the previous step, the code trims the audio by excluding the silent parts at the beginning and end. The resulting trimmed audio is stored in the "trimmed_sound" variable. Finally, the trimmed audio is exported as a new WAV file. The code saves the file in the "modified_files" directory using the original filename with "_trimmed" appended to it.
- **detect_leading_silence:** The "detect_leading_silence" function is a helper function used by "removeSilenceParts" to determine the duration of leading silence. The function takes an audio file as input along with optional parameters such as the silence threshold and chunk size. The code checks the silence level of the audio before the first chunk using the specified silence threshold. If the silence level is above the threshold, indicating the absence of leading silence, the function returns 0. The function iterates over small chunks of audio until it finds the first chunk that exceeds the specified silence threshold, indicating the presence of sound. It keeps track of the accumulated time in milliseconds (trim_ms) to determine the duration of the leading silence. The function returns the value of trim_ms, representing the duration of the leading silence in the audio file.

The implementation of this procedure is given particular attention, as it exclusively involves the selection of the initial and final segments of the audio files for trimming purposes. The retention of potential silence in the middle sections of a recording is of utmost importance, as the elimination of initial and trailing silence serves distinct purposes, such as facilitating the comprehension of words with numerous compounds.

```python
def removeSilenceParts(audio,name):

    start_trim = detect_leading_silence(audio)
    end_trim = detect_leading_silence(audio.reverse())

    duration = len(audio)
    trimmed_sound = audio[start_trim:duration-end_trim]
    # Export the trimmed sound to a WAV file

    trimmed_sound.export("./modified_files/" + name + "_trimmed.wav", format="wav")

def detect_leading_silence(sound, silence_threshold=-50.0, chunk_size=10):
    trim_ms = 0 # ms

    assert chunk_size > 0 # to avoid infinite loop

    # Check the silence level of the audio before the first chunk
    pre_chunk_silence = sound[:chunk_size].dBFS
    if pre_chunk_silence >= silence_threshold:
        return 0

    # Iterate over chunks until you find the first one with sound
    while sound[trim_ms:trim_ms+chunk_size].dBFS < silence_threshold and trim_ms < len(sound):
        trim_ms += chunk_size

    return trim_ms
```

Listing 6: Python code: Removing the silence parts

6 Evaluation

The objective of this chapter is to assess and choose the optimal strategies for addressing the difficulties presented by the four essential parameters ranging from CHA-1 to CHA-4. The challenges discussed in the previous chapters have been comprehensively analysed and viable solutions have been proposed. The present discourse centres on a thorough assessment of the aforementioned proposed tactics. The principal aim is to evaluate the compromises linked with every proposed resolution and appraise their pragmatic ramifications. The objective at the conclusion of the chapter is to ascertain and propose the most suitable methods of mitigation that exhibit the highest capacity to tackle the aforementioned parameters. The ultimate objective is to augment the comprehensive calibre and precision of ASR systems.

6.1 Evaluation of the parameter CHA-1

In the context of ASR, the trade-offs between noise reduction and oversmoothing techniques need to be carefully considered to enhance the accuracy and reliability of the speech recognition system. While both techniques aim to improve the quality of the audio input, their impact on ASR performance differs. The main objective of noise reduction methods is to address the issue of ambient noise, which poses a considerable obstacle in the context of ASR. The objective of these techniques is to improve the clarity and comprehensibility of speech by attenuating extraneous noise, thereby augmenting the signal-to-noise ratio. It is crucial to recognise that an overabundance of noise reduction techniques may unintentionally eliminate significant speech characteristics, ultimately resulting in the forfeiture of crucial information required for precise speech recognition. Addition-

ally, aggressive noise reduction can introduce artifacts or distortions that could negatively impact the ASR system's performance.

Conversely, oversmoothing methodologies aim to mitigate sudden variations and fluctuations in the auditory signal. While oversmoothing can help improve the visual appearance of the signal and reduce high-frequency noise, it can also cause a loss of fine-grained details and distort the original speech characteristics. The loss of such details may negatively affect the ASR system's ability to accurately capture phonetic nuances, resulting in decreased recognition accuracy and potential misinterpretation of speech.

Persona Name	**Limitations**
Reduces background noise interference, thereby improving speech intelligibility.	Important speech details and acoustic cues may be lost.
Increased signal-to-noise ratio, reducing noise-induced errors.	Overreliance on noise reduction may result in speech signal distortion or alteration.
Improved speech recognition accuracy, particularly in noisy environments.	Inaccurate noise estimation or overzealous noise reduction may result in the introduction of artefacts.
Improved speech-to-noise separation improves ASR performance.	Insufficient noise suppression, resulting in noise-related ASR errors.
-	The presence of residual noise or artefacts in the processed audio, which inhibits the performance of the ASR.
-	Due to ineffective noise reduction, speech characteristics become distorted or unnatural.
-	Inefficient algorithms for noise reduction may increase computational complexity.

Table 2: Advantages and Limitations of noise reduction techniques for ASR.

Advantages	**Limitations**
Potential reduction of background noise interference.	Loss of important speech details and acoustic cues.
Suppression of certain types of noise, such as stationary or continuous background noise.	Distortion or alteration of speech signal, leading to reduced intelligibility.
Reduction of overall noise level, which may enhance speech-to-noise ratio.	Potential introduction of artifacts or unnatural speech characteristics.
Improved speech-to-noise separation improves ASR performance.	Insufficient noise suppression, resulting in noise-related ASR errors.
-	Oversmoothing can lead to the loss of critical speech features, negatively impacting ASR accuracy.
-	Potential reduction in speech intelligibility due to the removal of important speech cues.
-	Introduction of unnatural or distorted speech characteristics, affecting ASR system performance.

Table 3: Advantages and Limitations of oversmoothing techniques for ASR.

Given the trade-offs involved, as shown in Table 2 and Table 3, it is recommended that, in the particular scenario under consideration, the ASR system would be **better served by employing**

noise reduction techniques as opposed to oversmoothing. Although oversmoothing may enhance the signal's visual appeal, noise reduction directly tackles the main obstacle of background noise, which has a substantial impact on the performance of ASR. The ASR system can improve its overall accuracy by enhancing the signal-to-noise ratio through a focus on noise reduction, which in turn allows for better recognition of speech patterns.

Thus, considering the aim of improving ASR accuracy while mitigating the influence of ambient noise, the choice to prioritise noise reduction methods over excessive smoothing is rational. This decision guarantees that the ASR system can proficiently manage audio inputs from real-world scenarios that exhibit diverse levels of noise, leading to more dependable and precise speech recognition results.

6.2 Evaluation of the parameter CHA-2

In the realm of ASR, it is crucial to assess the benefits and drawbacks of speaker variability training sets and speaker normalisation techniques when making trade-offs. Both methodologies provide unique advantages that can substantially improve the efficiency of an ASR system.

The utilisation of a speaker variability training set confers various benefits. The ASR system can attain greater robustness and adaptability to diverse speaking styles and variations by subjecting it to a wide range of speakers during the training process. The aforementioned capability allows the system to exhibit a high degree of generalisation towards speakers who were not previously encountered, thereby mitigating the influence of speaker-dependent biases and enhancing the overall precision of the system. Moreover, the training set for speaker variability promotes speaker autonomy, thereby enhancing the ASR system's ability to effectively manage real-life situations characterised by significant variations in speakers' attributes.

Conversely, techniques for speaker normalisation present a distinct array of benefits. The objective is to minimise the impact of individual speaker traits on the ASR system by lessening the speaker-dependent variances. Through the process of speech signal normalisation, these techniques improve the system's capacity to model and identify speech patterns that are not limited to individual speakers, thereby facilitating generalisation to speakers who have not been previously encountered. The normalisation of speaker characteristics is a useful technique for comparing and analysing speech data from multiple speakers, rendering it advantageous in diverse applications.

Although both methodologies possess their own merits, it is imperative to recognise their respective constraints. The process of obtaining a training set for speaker variability necessitates a diverse and representative group of speakers, which can be a challenging and resource-intensive task. The efficacy of speaker normalisation methods is contingent upon precise assessment of speaker attributes, which may not always be attainable, and there exists a conceivable hazard of introducing anomalies or deformations in the speech signal while executing the normalisation procedure.

Advantages	Limitations
Improved robustness to speaker-dependent variations.	Requires a diverse and representative set of speakers for training, which can be resource-intensive.
Better generalization to unseen speakers.	Difficulty in capturing all possible speaker variations.
Increased adaptability to different speaking styles.	Speaker variability alone may not address all sources of variation in the speech signal.
Reduces speaker-specific biases.	Potential risk of overfitting to training speakers.

Table 4: Advantages and Limitations of speaker variability training set.

Given the manifold benefits afforded by the **utilisation of both speaker variability training sets and speaker normalisation techniques**, as shown in Table 4 and Table 5, it is advisable to integrate both methodologies into the project. The collaborative utilisation of their respective

Advantages	Limitations
Reduces speaker-dependent variations.	Speaker normalization techniques may introduce artifacts or distortions in the speech signal.
Helps improve speaker-independent modeling.	Requires accurate estimation of speaker characteristics, which may not always be feasible.
Enhances generalization to unseen speakers.	Incomplete or inaccurate normalization can lead to loss of important speaker-specific information.
Mitigates the influence of speaker variability.	Normalization techniques may not fully eliminate all sources of variation, such as pronunciation errors.

Table 5: Advantages and Limitations of techniques for normalization techniques for ASR.

strengths can lead to a synergistic enhancement of the performance of the ASR system. Through the utilisation of speaker variability training set, the system can attain enhanced resilience, flexibility, and diminished speaker partialities. The utilisation of speaker normalisation techniques concurrently aids in the reduction of speaker-dependent variations, amplification of generalisation, and simplification of comparison among speakers.

The project aims to enhance the reliability and accuracy of the ASR system by incorporating both speaker variability training set and speaker normalisation techniques. This integration allows the project to leverage the benefits of each approach. The integration of these components facilitates the efficient management of diverse speech patterns, adaptation to speaker idiosyncrasies, and enhancement of global efficacy, in accordance with the objectives of the study to attain superior speech recognition results.

6.3 Evaluation of the parameter CHA-3

In the realm of ASR, it is imperative to assess the benefits and drawbacks of reverberation and noise reduction methods, as well as their potential efficacy in addressing CHA-3. Both methodologies possess the capability to tackle environmental variables such as ambient noise and sound reflection, however, their appropriateness for ASR necessitates a meticulous evaluation.

The implementation of techniques aimed at reducing reverberation presents benefits in terms of minimizing the influence of the acoustic properties of a room and ameliorating the deterioration resulting from the presence of echoes. The implementation of these techniques can potentially enhance speech intelligibility and alleviate the negative impact of environmental factors by minimizing reverberation. The precise determination of the Room Impulse Response (RIR) can pose difficulties, and the efficacy of reducing reverberation is contingent upon the particular attributes of the room. Moreover, there exists a potential hazard of introducing artifacts or distortions in the processed speech, which may have an adverse impact on the accuracy of ASR. In light of the constraints identified and their consequential influence on the accuracy of ASR, the project team has made the determination to abstain from implementing reverberation reduction methods in the project.

In contrast, the implementation of noise reduction techniques presents notable benefits in ameliorating the influence of ambient noise, a prominent environmental variable that detrimentally affects ASR efficacy. The implementation of these techniques results in an improvement of speech intelligibility and a reduction of noise, leading to a more precise and refined speech signal, as shown in Table 2. As previously stated, in subsection 6.2, the project team has opted to incorporate noise reduction methodologies into the project's workflow. The aforementioned decision is consistent with the project's aim of enhancing ASR precision through the reduction of the impact of ambient noise.

To sum up, the utilization of both reverberation and noise reduction techniques exhibits the capability to alleviate environmental factors in ASR. Although reverberation reduction techniques

Advantages	Limitations
Improved robustness to reverberant environments.	Accurate estimation of RIR can be challenging.
Enhances speech intelligibility in reverberant spaces.	The effectiveness of reverberation reduction depends on the specific room characteristics.
Helps minimize the impact of room acoustics.	Reverberation reduction techniques may introduce artifacts or distortions in the processed speech.
Mitigates the degradation caused by echoes.	Highly reverberant environments may still pose challenges for accurate speech recognition, despite reduction.

Table 6: Advantages and Limitations of reverberation technique for ASR.

have benefits in managing reverberation and room acoustics, their constraints and probable influence on ASR precision have prompted the project team to refrain from utilizing them. By way of comparison, the implementation of noise reduction techniques yields substantial advantages in ameliorating ambient noise, a pivotal environmental variable that impacts the efficacy of ASR. Hence, the implementation strategy **prioritizes the reduction of noise** as a dependable method for enhancing the precision of ASR in the face of environmental variables.

6.4 Evaluation of the parameter CHA-4

The investigation conducted by the project pertaining to the reduction of pronunciation errors has uncovered a number of efficacious methodologies, such as speech enhancement, the utilization of language models, and the incorporation of diverse speakers, as expounded upon in preceding sections. The employment of these methodologies in conjunction serves to effectively tackle the issue of mispronunciation in ASR systems.

The utilization of speech enhancement techniques presents benefits in the enhancement of speech intelligibility and the amplification of the signal-to-noise ratio. The process of speech enhancement is of paramount importance in mitigating the influence of environmental factors on the precision of pronunciation. This is achieved through the reduction of background noise and the minimization of distortion and artifacts. It is crucial to take into account the constraints related to the precise differentiation of speech from ambient noise and the probable forfeiture of significant speech data while enhancing it. Furthermore, it is important to consider the computational complexity and potential latency that may arise from the implementation of speech enhancement algorithms.

The utilization of language models for the purpose of reducing pronunciation errors yields considerable advantages. Language models can enhance the contextual relevance of the ASR system by improving its accuracy in handling pronunciation variations, recognizing words with variations, and better handling regional accents and dialects. The careful consideration of the availability and quality of training data is imperative for the language model. It is plausible that valid pronunciations may be overcorrected or unfamiliar/non-standard pronunciations may be misinterpreted. Moreover, the incorporation and application of the linguistic model may necessitate augmented computational capabilities.

Moreover, the project acknowledges the noteworthy influence of employing diverse speakers to tackle pronunciation inaccuracies. As elucidated in preceding sections, the incorporation of varied speech patterns, accents, and dialects facilitates the conditioning of the ASR mechanism to identify and comprehend a broader spectrum of pronunciations. This methodology improves the resilience and versatility of the system in accommodating diverse speech patterns.

After careful consideration of the trade-offs and time constraints associated with the project, the team has opted to integrate **multiple speakers**, as previously discussed in subsection 6.2, and **employ language models** as the ultimate strategies. The aforementioned decisions were made after a thorough evaluation of the benefits of utilizing speech enhancement techniques to rectify

Advantages	Limitations
Improved speech intelligibility.	Difficulty in accurately separating speech from background noise or interference.
Enhanced signal-to-noise ratio.	Potential loss of important speech information during the enhancement process.
Reduction of background noise.	Sensitivity to the quality of the input audio and the specific characteristics of the noise.
Minimization of distortion and artifacts.	Computational complexity and potential latency introduced by the speech enhancement algorithms.

Table 7: Advantages and Limitations of speech enhancement technique for ASR.

Advantages	Limitations
Improved speech intelligibility.	Difficulty in accurately separating speech from background noise or interference.
Enhanced signal-to-noise ratio.	Potential loss of important speech information during the enhancement process.
Reduction of background noise.	Sensitivity to the quality of the input audio and the specific characteristics of the noise.
Minimization of distortion and artifacts.	Computational complexity and potential latency introduced by the speech enhancement algorithms.

Table 8: Advantages and Limitations of using language models.

pronunciation errors, while also taking into account the potential obstacles and constraints that may arise. The project endeavors to enhance the performance of the ASR system and provide precise outcomes in diverse pronunciation scenarios by utilizing the variety of speakers and exploiting the contextual comprehension offered by language models.

6.5 Comprehensive Evaluation of ALL Challenges

This subsection provides a thorough assessment of various methods utilized for the purpose of ASR. The objective is to evaluate the efficacy of said techniques in achieving diverse evaluation objectives within the framework of ASR systems. The assessment comprises crucial facets such as comprehensibility of speech, existence of ambient noise, elimination of significant speech data, general enhancement in audio caliber, cost-benefit ratio, simplicity of integration and utilization, and duration of audio processing. The objective is to offer a comprehensive perspective on the methodologies and their efficacy in augmenting ASR capabilities. This assessment facilitates the process of making well-informed decisions pertaining to the choice and execution of suitable methodologies, taking into account their efficacy in addressing prevalent obstacles in automatic speech recognition. This resource is highly valuable for individuals involved in research, practice, and development who aim to enhance the functionality and user-friendliness of ASR systems across diverse applications and settings.

Legend:
✓ (Single Tick): Indicates a positive outcome or effectiveness in relation to the corresponding evaluation objective.
✗ (Single Cross): Indicates a negative outcome or ineffectiveness in relation to the corresponding evaluation objective.
✓✓ (Double Tick): Represents a higher level of positive outcome or effectiveness compared to other techniques in relation to the corresponding evaluation objective.

✗✗ (Double Cross): Represents a higher level of negative outcome or ineffectiveness compared to other techniques in relation to the corresponding evaluation objective.

The evaluation metrics chosen for assessing the performance of the audio cleansing techniques

ID	Name of the technique
CT-1.	Remove Background Noise.
CT-2.	Remove Silence Parts.
CT-3.	Normalization.
CT-4.	Compression.
CT-5.	Use of Variety of Speakers.
CT-6.	Over-smoothing.
CT-7.	Reverberation.
CT-8.	Speech Enhancement.
CL-9.	Employ Language Models.

Table 9: Identifiers of various Cleaning Techniques(CT).

were meticulously chosen to address key aspects pertinent to the ASR system. Each evaluation metric serves a distinct function in assessing the efficacy and suitability of the techniques for ASR applications.

- **Understandability of Speech:** This metric seeks to evaluate the clarity and intelligibility of the speech after the cleaning techniques have been applied. It ensures that the processed audio remains understandable, allowing the ASR system to accurately transcribe the speech. A checkmark indicates enhanced speech clarity and intelligibility, whereas a cross indicates a detrimental effect on speech comprehension.
- **Presence of Background Noise:** Background noise can degrade the efficacy of ASR systems significantly. Therefore, evaluating the efficacy of noise reduction techniques is essential for minimizing the impact of background noise and enhancing the overall quality of speech. A checkmark indicates that external noise has been effectively reduced or eliminated, whereas a cross indicates that noise reduction has been ineffective.
- **Removal of Important Speech Information:** When removing noise, it is essential to ensure that essential speech information is not eliminated or distorted unintentionally. This metric is used to evaluate a technique's ability to eradicate noise selectively while preserving crucial speech details. A checkmark indicates the effective preservation of crucial speech details, whereas a cross indicates the unintentional removal of such details.
- **Overall Improvement in Audio Quality:** This metric evaluates the overall improvement in audio quality brought about by the cleaning techniques, taking into account factors such as the speech's intelligibility, fidelity, and naturalness. A checkmark denotes an improvement in audio quality across the board, including clarity, intelligibility, fidelity, and naturalness. A cross signifies a deterioration in audio quality.
- **Cost-effectiveness of the technique:** Important considerations for "cost-effectiveness of the technique" include implementation feasibility and resource requirements. Evaluation of the cost-effectiveness metric identifies techniques that establish a balance between computational complexity, time efficiency, and financial costs, ensuring their applicability in real-world situations. Tick is a solution that is efficient in terms of computational resources, time, and financial expenditures. A cross represents a less cost-effective strategy.
- **Ease of Implementation and Use:** Usability and ease of implementation are crucial factors in the selection of cleaning techniques. This metric enables us to identify techniques

that are simple to integrate into existing systems and require minimal user input. The checkmark denotes the technique's simple implementation and user-friendliness. A cross signifies implementation or usage difficulties or complexities.

- **Time Required to Process the Audio:** Processing time plays a crucial role in real-time applications. This metric assesses the effectiveness and speed of audio file cleansing techniques, ensuring that the selected techniques can process audio within acceptable time constraints. A checkmark indicates a quick and efficient processing time, whereas a cross indicates lengthier processing times or inefficiency in audio file management.

By selecting these specific evaluation metrics, we ensure a comprehensive assessment of the performance of the cleaning techniques in the context of ASR, taking into account various critical aspects such as speech understandability, noise reduction, speech information preservation, overall audio quality, cost-effectiveness, ease of implementation, and processing efficiency.
Table 10 presents a comprehensive tabular comparison of diverse evaluation techniques for ASR within the framework of distinct cleaning techniques. The aforementioned table presents a comprehensive summary of the efficacy of individual techniques in tackling primary evaluation objectives. The assessment methodologies comprise a total of seven and encompass a wide spectrum of facets that are pertinent to ASR systems. The tabular format facilitates a straightforward evaluation of the methodologies, revealing their respective efficacy. This thorough assessment facilitates informed decision-making when choosing the most appropriate methods for enhancing ASR accuracy in diverse contexts.

Evaluation objectives	**CT-1**	**CT-2**	**CT-3**	**CT-4**	**CT-5**	**CT-6**	**CT-7**	**CT-8**	**CT-9**
Understandability of speech	✓✓	✓	✓✓	✓	✓✓	✗✗	✓✓	✓✓	✓✓
Presence of background noise	✓✓	✓	✓	✗✗	✓✓	✗✗	✓	✓✓	✓✓
Removal of important speech information	✗✗	✗✗	✗✗	✗✗	✓✓	✓✓	✗✗	✓	✓✓
Overall improvement in audio quality	✓✓	✗✗	✓✓	✓✓	✓✓	✗✗	✓	✓✓	✓✓
Cost-effectiveness of the technique	✓✓	✓✓	✓	✓	✓✓	✓	✗✗	✓✓	✓✓
Ease of implementation and use	✓✓	✓✓	✓	✓	✓✓	✓	✗✗	✓✓	✓✓
Time required to process the audio	✓✓	✓✓	✓	✓	✓✓	✗✗	✗✗	✓	✓✓

Table 10: Evaluation of ASR Cleaning Techniques

The present illustration showcases the process of interpreting the evaluation table and acquiring valuable insights regarding the efficacy of individual techniques in mitigating background noise to enhance ASR performance. Various methods were examined to determine their efficacy in reducing the impact of ambient noise during the assessment of its existence. The assessment of subsections 6.1, 6.2, 6.3, and 6.4 revealed that the methods of noise reduction and speech enhancement exhibited favorable results in mitigating the influence of ambient noise, as evidenced by a solitary checkmark in the respective cells of the table. In terms of efficacy, speech enhancement exhibited a superior level of achievement in mitigating ambient noise, as indicated by the presence of two check marks in its corresponding cell. This indicates that speech enhancement yielded superior results compared to noise reduction in mitigating the effects of ambient noise. Alternative methodologies, such as incorporating diverse speakers and utilizing linguistic models, demonstrated a degree of

efficacy, albeit not comparable to that of speech enhancement. On the other hand, it was observed that methods such as over-smoothing and dereverberation exhibited lower efficacy in reducing the impact of ambient noise, as denoted by the symbol of a single cross in the corresponding cells. In general, the assessment underscores the differing levels of efficacy exhibited by distinct approaches in mitigating the impact of ambient noise, with speech enhancement being identified as the most efficacious technique in this regard.

6.6 Implementing Combined Audio Cleaning Techniques

The Python code presented in Listing 7 includes a function named clean_audio_files that is designed to enhance the quality of audio files utilized in automatic speech recognition machine learning models. The program sequentially traverses a collection of filenames denoted as metadata and executes a set of procedures aimed at enhancing the quality of each audio file. Subsequently, the code executes a series of cleansing methodologies on the audio file with the objective of improving its appropriateness for automated speech recognition. The aforementioned techniques are chosen based on the assessment of the preceding subsections. The code incorporates a visualization step that offers valuable insights into the impact of individual cleaning techniques on the audio file. The code facilitates a more comprehensive comprehension of the enhancements attained through the employed cleansing methodologies by means of visualizing the audio files at various phases.

```
def clean_audio_files(metadata):
    directory = "./downloaded_files/"
    for item in metadata:
        full_path = directory + item
        if not os.path.exists(full_path):
            print(f"File {full_path} does not exist. Skipping...")
            continue
        file_name_with_extension = os.path.basename(full_path)
        file_name_without_extension = os.path.splitext(file_name_with_extension)[0]
        sound = AudioSegment.from_file(full_path)
        print(full_path)
        visualization(sound,file_name_without_extension,"dB","Original audio file")
        noice_reduction(sound,file_name_without_extension)
        sound_noice_reduction = AudioSegment.from_file("./modified_files/" +
    file_name_without_extension + "_noice_red"+".wav")
        visualization(sound_noice_reduction,file_name_without_extension + "
    _noice_red", "dB", "Audio file after noise reduction")
        removeSilenceParts(sound_noice_reduction,file_name_without_extension)
        sound_trimmed = AudioSegment.from_file("./modified_files/" +
    file_name_without_extension + "_trimmed"+".wav")
        visualization(sound_trimmed,file_name_without_extension + "_trimmed", "dB",
    "Audio file after removing silence parts")
        normalization(sound_trimmed, file_name_without_extension)
        sound_normalized = AudioSegment.from_file("./modified_files/" +
    file_name_without_extension + "_normalized"+".wav")
        visualization(sound_normalized,file_name_without_extension + "_normalized",
     "dB", "Audio file after normalization")
        oversmoothing(sound_trimmed,file_name_without_extension, window_size=5)
        sound_overmoothing = AudioSegment.from_file("./modified_files/" +
    file_name_without_extension + "_overmoothing"+".wav")
        visualization(sound_overmoothing,file_name_without_extension + "
    _overmoothing", "dB", "Audio file after overmoothing")
```

Listing 7: Python code: Final cleaning techniques

7 Results and Discussion

7.1 Factors Affecting ASR Accuracy

The first research question **(RQ1)** aims to identify the primary factors that significantly affect the accuracy of ASR and explores effective strategies to mitigate their impact. The investigation has revealed that various crucial factors have been identified, such as **background noise, speaker**

variability, environmental factor, pronunciation errors. The aforementioned factors have been extensively acknowledged as significant obstacles in attaining a high degree of accuracy in ASR.

The presence of background noise presents a notable hindrance to precise speech recognition. In order to tackle this issue, various methods for reducing noise have been widely utilized. Through the implementation of noise suppression techniques and the enhancement of the signal-to-noise ratio, the capacity of the ASR system to identify speech patterns is augmented, leading to a subsequent improvement in accuracy.

The performance of ASR is significantly influenced by the factor of speaker variability. In order to reduce its impact, two primary strategies have been implemented, namely, the utilization of training sets that incorporate speaker variability and the application of techniques for speaker normalization. Training sets that incorporate speaker variability expose the ASR system to a range of speakers with distinct speech patterns, speaking styles, and individual characteristics. This enables the system to acquire knowledge and adjust to diverse speech inputs. The implementation of speaker normalization techniques serves to mitigate the influence of speaker-specific variations, thereby fostering equitable comparisons across speakers and augmenting the capacity for generalization.

ASR accuracy can be influenced by environmental factors, including reverberation. Although various methods for reducing reverberation have been explored, their impact on the accuracy of ASR has prompted us to refrain from incorporating them into our project. Our research has primarily centered on the implementation of noise reduction techniques. These techniques have been found to be effective in mitigating the issue of ambient noise, thereby leading to a substantial enhancement in the accuracy of ASR systems.

ASR systems encounter an additional obstacle in the form of inaccuracies in pronunciation. The employment of language models has demonstrated significant efficacy in reducing their impact. The accuracy of ASR can be enhanced by integrating contextual comprehension, grammar, and vocabulary knowledge into language models, which facilitate precise word recognition.

7.2 Advantages and Trade-Offs of Audio Cleaning Techniques

The second research question **(RQ2)** delves into the advantages and trade-offs of employing various combinations of audio cleaning techniques in the preparation of audio files for machine learning models used in automatic speech recognition. The aim is to ascertain the optimal cleaning methodologies that can enhance the ultimate outcomes of the ASR system.

By conducting a comprehensive assessment and juxtaposition of diverse cleaning methodologies, encompassing noise reduction, speaker variability training sets, speaker normalization, the incorporation of multiple speakers, and language models, the benefits and drawbacks of each approach were scrutinized.

The chosen cleaning techniques have shown notable benefits in effectively tackling particular difficulties. The implementation of noise reduction techniques has been observed to have a substantial impact on the signal-to-noise ratio, thereby resulting in the enhancement of speech pattern recognition and improved accuracy of ASR systems. The utilization of speaker variability training sets and speaker normalization techniques has been found to promote adaptability, versatility, and decreased speaker biases, while also decreasing speaker-specific variations and enhancing the ability to generalize.

The incorporation of multiple speakers has demonstrated significant advantages in managing a wide range of accents, dialects, and speech patterns, leading to a more resilient and versatile ASR mechanism. Finally, the integration of linguistic models offers contextual comprehension, grammatical accuracy, and lexical proficiency, thereby augmenting the precision of ASR systems.

Nevertheless, there are trade-offs associated with the utilization of these techniques. Reverberation reduction techniques, although advantageous in the realm of room acoustics management, impose certain constraints that may have an adverse effect on the precision of ASR. In light of the trade-offs involved and a thorough assessment of temporal limitations, our project has made the decision to abstain from employing reverberation reduction techniques.

To sum up, the utilization of a blend of methods for reducing noise, training sets for speaker variability, techniques for normalizing speakers, the incorporation of diverse speakers, and language

models results in significant advantages for enhancing the precision and efficacy of automatic speech recognition systems. The aforementioned techniques are efficacious in mitigating crucial factors that exert a substantial impact on the accuracy of ASR systems, such as ambient noise, speaker diversity, and enunciation inaccuracies. Through the reduction of the influence of these variables and the optimization of the advantages of each method, the ASR system can attain greater precision, increased robustness, and improved recognition aptitudes, resulting in more accurate results across a range of pronunciation contexts.

8 Conclusion and Learnings

The present investigation has examined the variables that exert a significant influence on the precision of ASR and has probed into potential approaches to alleviate their effects. The study involved a comprehensive examination of existing literature, meticulous scrutiny of data, experimentation, and coding. This endeavor yielded significant insights and knowledge, encompassing both the research inquiries tackled and the research methodology as a whole.

The literature review furnished a thorough comprehension of ASR and audio cleaning methodologies, pinpointing areas of research that require further exploration and laying a strong groundwork for the inquiry. The activity refined the individual's ability to engage in critical thinking and analysis, by assessing the merits and limitations of diverse research studies.

The phase of coding and experimentation has augmented the programming proficiency and provided insights into the complexities of dealing with actual audio data in practical scenarios. The researchers acquired the skills to identify and resolve problems, optimize variables, and analyze findings from experiments. The significance of rigorous experimental design, precise data management, and the necessity for versatility and adjustability was acknowledged by them.

The utilization of collaboration and interdisciplinary perspectives was found to be highly valuable, as it facilitated the expansion of knowledge and the cultivation of novel concepts. The process of research was enriched through active participation in discussions and conferences with peers and experts.

The study has identified key factors that impact the accuracy of ASR systems, including but not limited to background noise, speaker variability, environmental factors, and pronunciation errors. These factors were found to be significant in addressing the research questions. The study revealed that the implementation of noise reduction methods, speaker variability training sets, speaker normalization techniques, the integration of multiple speakers, and language models had a notable positive impact on the accuracy of ASR.

In summary, this research endeavor facilitated an enhanced comprehension of the obstacles encountered in ASR, honed technical competencies, and cultivated a sophisticated viewpoint on audio purification and machine learning in ASR implementations. The aforementioned experiences underscored the significance of careful and thorough planning, rigorous assessment of information, flexibility, and cross-disciplinary partnerships.

The acquired knowledge and insights have implications that transcend the confines of the particular research inquiries, providing a navigational reference point for forthcoming scholarly undertakings. The study's shared insights have the objective of propelling ASR research, stimulating additional inquiries, and making a contribution to the dynamic and evolving domain of automatic speech recognition.

List of Acronyms

ASR Automatic Speech Recognition

CMN Cepstral Mean Normalisation

CNN Convolutional Neural Network

HMM Hidden Markov Model

DSP Digital Signal Processing

fMLLR Feature Space Maximum Likelihood Linear Regression

GAN Generative Adversarial Networks

ISTFT Inverse Short-Time Fourier Transform

LSTM Long Short-Term Memory

OOV out-of-vocabulary

RIR Room Impulse Response

RNN Recurrent Neural Networks

RQ Research Question

SNR Signal-to-Noise Ratio

STFT Short-Time Fourier Transform

STFT Short-Time Fourier Transform

TDNN Time-Delay Neural Networks

WAV Waveform Audio File Format

3gp Third Generation for mobile Platform

References

[1] S. B. Davis, R. Biddulph, and S. Balashek. "Automatic recognition of spoken digits". In: *The Journal of the Acoustical Society of America* 24.6 (1952), pp. 637–642.

[2] J. K. Baker. "The dragon system—An overview". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23.1 (1975), pp. 24–29.

[3] H. Bourlard and N. Morgan. *Connectionist speech recognition: A hybrid approach*. Kluwer Academic Publishers, 1994.

[4] G. Hinton et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97.

[5] A. Y. Hannun et al. "Transfer learning from speaker verification to multispeaker text-to-speech synthesis". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 2019, pp. 6965–6969.

[6] O. Abdel-Hamid et al. "Convolutional neural networks for speech recognition". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.10 (2014), pp. 1533–1545.

[7] A. Y. Hannun et al. "Deep speech: Scaling up end-to-end speech recognition". In: *arXiv preprint arXiv:1412.5567* (2014).

[8] Gibak Kim and Philipos Loizou. "Improving Speech Intelligibility in Noise Using Environment-Optimized Algorithms". In: *Audio, Speech, and Language Processing, IEEE Transactions on* 18 (Dec. 2010), pp. 2080–2090. DOI: 10.1109/TASL.2010.2041116.

[9] Krishnamoorthy Palanisamy and S. Prasanna. "Enhancement of noisy speech by temporal and spectral processing". In: *Speech communication* 53 (Feb. 2011), pp. 154–174. DOI: 10.1016/j.specom.2010.08.011.

[10] Dayana Ribas et al. "Wiener Filter and Deep Neural Networks: A Well-Balanced Pair for Speech Enhancement". In: *Applied Sciences* 12.18 (2022), p. 9000.

[11] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. "A time delay neural network architecture for efficient modeling of long temporal contexts". In: Sept. 2015, pp. 3214–3218. DOI: 10.21437/Interspeech.2015-647.

[12] Sebastian Braun and Hannes Gamper. "Effect of Noise Suppression Losses on Speech Distortion and ASR Performance". en. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Singapore, Singapore: IEEE, May 2022, pp. 996–1000. ISBN: 978-1-66540-540-9. DOI: 10.1109/ICASSP43922.2022.9746489. URL: https://ieeexplore.ieee.org/document/9746489/.

[13] Zixing Zhang et al. "Deep learning for environmentally robust speech recognition: An overview of recent developments". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 9.5 (2018), pp. 1–28.

[14] Hiroshi Sato et al. "Should we always separate?: Switching between enhanced and observed signals for overlapping speech recognition". In: *arXiv preprint arXiv:2106.00949* (2021).

[15] Lalit Kumar and Dushyant Kumar Singh. "A comprehensive survey on generative adversarial networks used for synthesizing multimedia content". In: *Multimedia Tools and Applications* (2023), pp. 1–40.

[16] Sung Kim and Visvesh Sathe. "Adversarial audio super-resolution with unsupervised feature losses". In: (2018).

[17] Nasser Mohammadiha, Paris Smaragdis, and Arne Leijon. "Supervised and unsupervised speech enhancement using nonnegative matrix factorization". In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.10 (2013), pp. 2140–2151.

[18] L. P. García-Perera, J. Lorenzo-Trueba, and A. G. Álvarez-Marquina. "Speech recognition in noisy environments: An overview of the problem and the solutions". In: *Applied Sciences* 10.16 (2020), p. 5585. DOI: 10.3390/app10165585. URL: https://www.mdpi.com/2076-3417/10/16/5585.

[19] Brian Kingsbury et al. "Robust speech recognition in noisy environments: the 2001 IBM SPINEevaluation system". In: vol. 1. Feb. 2002, pp. I–53. ISBN: 0-7803-7402-9. DOI: 10.1109/ICASSP.2002.5743652.

[20] Hans-Günter Hirsch and David Pearce. "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions". In: *ASR2000-Automatic speech recognition: challenges for the new Millenium ISCA tutorial and research workshop (ITRW)*. 2000.

[21] Kunnar Kukk and Tanel Alumäe. "Improving Language Identification of Accented Speech". In: *arXiv preprint arXiv:2203.16972* (2022).

[22] K. Livescu and J. Glass. "Lexical modeling of non-native speech for automatic speech recognition". In: *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*. Vol. 3. 2000, 1683–1686 vol.3. DOI: 10.1109/ICASSP.2000.862074.

[23] Nina Markl. "Language Variation and Algorithmic Bias: Understanding Algorithmic Bias in British English Automatic Speech Recognition". In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '22. Seoul, Republic of Korea: Association for Computing Machinery, 2022, pp. 521–534. ISBN: 9781450393522. DOI: 10.1145/3531146.3533117. URL: https://doi.org/10.1145/3531146.3533117.

[24] Amalia Arvaniti. "Linguistic practices in Cyprus and the emergence of Cypriot Standard Greek". In: *Mediterranean Language Review* 17 (2010), pp. 15–45.

[25] Đorđe Grozdić and Slobodan Jovicic. "Whispered Speech Recognition Using Deep Denoising Autoencoder and Inverse Filtering". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (Dec. 2017), pp. 2313–2322. DOI: 10.1109/TASLP.2017.2738559.

[26] Petr Zelinka, Milan Sigmund, and Jiri Schimmel. "Impact of vocal effort variability on automatic speech recognition". In: *Speech Communication* 54.6 (2012), pp. 732–742. ISSN: 0167-6393. DOI: https://doi.org/10.1016/j.specom.2012.01.002. URL: https://www.sciencedirect.com/science/article/pii/S016763931200009X.

[27] Jont Allen and David Berkley. "Image method for efficiently simulating small-room acoustics". In: *The Journal of the Acoustical Society of America* 65 (Apr. 1979), pp. 943–950. DOI: 10.1121/1.382599.

[28] Matthias Wölfel and John McDonough. *Distant speech recognition*. John Wiley & Sons, 2009, pp. 7–10.

[29] Samia El-Moneim et al. "Text-independent speaker recognition using LSTM-RNN and speech enhancement". In: *Multimedia Tools and Applications* 79 (Sept. 2020). DOI: 10.1007/s11042-019-08293-7.

[30] Dafydd Gibbon, Roger Moore, and Richard Winski. *Handbook of Standards and Resources for Spoken Language Systems: Spoken language characterisation*. en. Google-Books-ID: 8cxtWcsAk5MC. Walter de Gruyter, 1997. ISBN: 978-3-11-015734-5.

[31] S. Levinson et al. "Speaker independent phonetic transcription of fluent speech for large vocabulary speech recognition". In: (Jan. 1989), pp. 75–80. DOI: 10.3115/100964.100965.