How Reduction in Sample Frequency Hinders the Detection of Words

LUCIA ALONSO ARENAZA
Supervisor(s): HAYLEY HUNG, JOSE VARGAS QUIROS, CHIRAG RAMAN
EEMCS, Delft University of Technology, The Netherlands

June 19, 2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Living in a world where every single electronic device is online and interconnected, privacy is a growing concern. Finding the threshold where audio is unintelligible to transcription software is crucial when everything that we say can be recorded. Even if Automated Speech Recognition (ASR) is used in tools, such as Siri or Alexa, designed to ease daily tasks, it could also be used in malicious manners. ASR technology has not been around for too long and like any other new piece of technology, it still has many aspects that have not been looked into and are unknown to the public. This research paper addresses this knowledge gap by examining how sample frequency reduction affects word detection using current well-known transcription software technology such as Google's speech recognition software and Kaldi's toolkit. The behavior and performance of these two software pieces have been analyzed for different sample frequencies in the range from 300Hz to 44,1kHz.

1 Introduction

Language recognition software is present in personal devices such as mobiles and tablets, hands-free assistants for cars, or electronic home devices such as Alexa [1] and it has become so universal that it is almost present in every aspect of our lives. Speech-to-text transcription is an important section of the language recognition field. Having a text representation of speech offers a broad set of interesting applications. To name a few examples, this technology allows people with hearing or speech impairment to access live audio or digital contents through automatic subtitling or transcription [2, 3]. ASR is used daily, as is the case with multiple apps from the Microsoft Office suite, such as Word, PowerPoint, or Outlook, which come with a tool called "Dictate" which turns speech into text using speech-to-text transcription software [4].

With the blooming of technology, AI can be used to transcribe audio to text [5, 6]. In the context of this research, we have chosen Automatic Speech Recognition (ASR) software to perform speech-to-text transcriptions, instead of manual inspection which would be more tedious and time-consuming [7]. Thus, we have chosen ASR as the most appropriate method to see the effect that lowering the sample (or sampling) frequency has on hindering the detection of words in audio.

Even if speech-to-text is used to ease and improve the usability of technology, as the previous examples show, it can also be used spitefully [8]. Nowadays anyone who owns a smartphone or any other device with a microphone has a device capable of recording in hand. Whatever anyone says can be recorded [9] and then transcribed which is extremely dangerous if sensitive information about the speaker has been caught in the audio. Since plain text files occupy less space than audio files, as we will demonstrate, it could be beneficial, in cases where there is limited storage, to transcribe audio files and store them in text format. According to Li Wang, the average delivery speed in English is between 150 and 190 words per minute [10]. Let's suppose that someone recorded audio of one minute long, which takes up on average 0.75Mb if saved in mp3 format[1]. It will contain approximately 170 words. A plain text file (txt) with this amount of words will take up 0,0019Mb approximately[2]. This practice is widely used to gather information that can be analyzed more easily in text form [11].

Regarding this part of the field, we will attempt to discover if lowering the sampling frequency affects the detection of words and therefore the transcription of them. It is important to study the effects of lowering sample frequency for many reasons, the main one being privacy. If lowering the sampling frequency does indeed affect the detection of words and makes transcription harder, then we can gain some privacy, knowing that whatever is recorded, can't be later on exploited. ConfLab is a project that is currently in place to develop automated behavior analysis on low sample frequency audio files. It is an initiative of the Socially Perceptive Computing Lab at the Delft University of Technology [12]. One of many interests from the ConfLab Team was to learn how reducing the sampling frequency affects the detection of words. To address this knowledge gap, this research focuses on the analysis of how the reduction in sample frequency hinders the detection of words and affects the privacy of the speaker. More specifically, we will try to answer the following question:

How does the reduction in sample frequency hinder the detection of words?

Specifically, we will try to reply to this question by observing how Google's Speech Recognition Software and Kaldi's toolkit perform at lower frequencies in the range from 300Hz to 44,1kHz[3]. It is especially interesting for this research to see what these algorithms achieve when working with audio with a sampling frequency of 2kHz since this is the most important range regarding perceived intelligibility [14].

Following this introductory section, after talking about related work to this research and getting some background information about the methods or technologies involved in ASR, there will be a section explaining the methodology followed in this research. Then, we will explain the experimental setup and the results that the previous methodology obtained. After, we will look at the ethical aspects of this research followed by the discussion and conclusions. Finally, suggestions for future work are mentioned in the last section.

[1] Calculation done using Sound Devices Audio Calculator.

[2] The average length of words in English is 4.7 characters. In a txt file, each character takes 2 bytes of space in Unicode. Total space for 170 words (with 169 spaces in between) would be: 170 * 4.7 * 2 + 169 * 2 = 1936 bytes.

[3] Which is the most common sampling rate used for music CDs [13].

2 Background and Related Works

Although a full description of ASR technology is out of the scope of this document, some brief descriptions about how natural voice is converted into text and how this technology has evolved will aid in the understanding of the subsequent sections of this paper.

The ASR system processes the voice's audio signal in several steps in a pipeline to obtain a text representation of the speech detected. A simplified visual representation of this process can be found in Figure 1. From an input such as a natural voice, the first steps revolve around the analog to digital conversion (ADC) of this analog input signal. Once a digital representation of the sound has been derived, the next steps try to improve the input, by normalizing the volume, removing noise, etc. Continuing in this fashion, several processes are applied in a chain to extract phonemes, then words, and finally, sentences. The acoustic model maps a feature vector to the phoneme. From there, the combination of phonemes is matched to the most likely word in the phonetic alphabet [15].

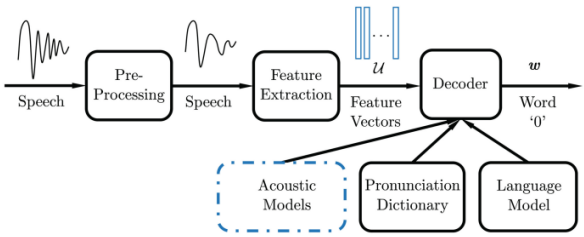

Figure 1: A simplified visual representation of the process ASR technology follows. Source: Graph by Kan Li [16, p. 3]

There are a couple of commercial ASR systems such as those by Google, Apple, etc., and some open-source systems; CMU Sphinx or Kaldi [17]. These systems have been trained using data that had its corresponding transcription. Open-source systems such as Kaldi offer the possibility of being adapted and trained with new data at will.

Something distinctive about this research is that the audio files have not been recorded in an ideal situation with no background noise or no other people speaking. On the contrary, it is done at a social gathering where the speech of several simultaneous speakers occurs. Additionally, the audio is in Dutch. This is not a problem but something to bear in mind to choose ASR systems that have been trained and have good accuracy in Dutch. In this aspect, research done by de Ruiter [18] shows that the Google Speech API software and Kaldi are the most accurate ASR systems to work with when using audio files in Dutch. Other characteristics that affect the precision of the transcriptions, apart from the language, are accents. Unfortunately, this aspect is still lacking behind in the world of ASR technology. "The difficulty is caused not only by deviations in pronunciation, but also by different vocabularies and even grammars that are used by speakers with different language and accent backgrounds" [19]. This aspect will be taken into account in section 6 when discussing the results of this research.

Many commonly applied techniques help the transcription process. For example, one technology that appears in the literature referring to transcription is Voice Activity Detection (VAD) technology. These types of software are used to "detect the presence or absence of speech in a segment of an acoustic signal" [20, p. 1]. The application of this process produces a result that indicates the occurrences of speech and silences in an audio file.

Similar to the previously mentioned initiative ConfLab, the MIT back in 2018 worked with an open-source wearable in the form of a smart badge, called RhythmBadge [21]. The aim of these badges was to record audio at low sample frequencies which in theory would not allow conversations to be intelligible. In the field of Social Studies, significant research has been conducted, such as ConfLab and RhythmBadge, devoted to studying social interactions without violating the privacy of the participants. The aim of this paper is to gain more knowledge about how frequency affects the detection of words, thus, furthering and adding knowledge to the field of social studies.

During this research, we are going to analyze the effect of different sampling frequencies working with two known ASR tools, combined with possible enhancements that could improve the process even with low sampling rates.

3 Methodology

We will begin by describing the structure and the audio characteristics of the dataset used for analysis in subsection 3.1. How audio files are processed is represented in subsection 3.2. This will include two subsections detailing methods and processes used in each of the categories; high and low-frequency audio. Then, a description of the ASR methods used will be in subsection 3.3. Finally, subsection 3.4 reflects on the methods used to evaluate the accuracy of the ASR methods.

3.1 The audio files: Layout and Information of the Dataset

It is of great importance to understand the layout and the specifics of the dataset before starting to talk about audio processing. In our case, we have been granted access to two different datasets that were collected by a group of researchers at TU Delft.

- **March15LaRedBirthdayParty**: contains audio files with a sample frequency of 44.1kHz.
- **Conflab-mm**: contains audio files with a sample frequency of 1.2kHz collected in the initiative ConfLab.

For this specific research, the dataset March15LaRedBirthdayParty was chosen. To compare the effect that lowering the sampling frequency has on privacy we need to know what was originally said in the audio files. In regards to Conflab-mm, the audios are unintelligible to the human ear, so there is no way to get

an original transcription for the audio files. For this reason, it was decided to use the March15LaRedBirthdayParty dataset.

March15LaRedBirthdayParty contains 16 different audio files that are 4 hours long. All of these audio files were recorded during a networking event. Each audio file corresponds to one speaker, there were 16 people wearing a microphone. Depending on the placement of the microphone, there are some audios that are more intelligible than others. In this event, there were no extraordinary measures taken to ensure the privacy of the speaker, the audio was recorded at a standard sample rate of 44.1kHz.

3.2 Pre-processing audio for ASR

Before starting processing any type of audio, a subset from the whole dataset needs to be selected. In this case, it was decided to work with one minute of each of the audio files where there were almost no silences. It was decided that it would be more beneficial to choose as many audio files as possible, instead of picking fewer files but with a longer duration, to account for different accents, speaking patterns, etc. Further motivation that supports this decision is described in subsection 3.3. To get this subset one minute where the speaker was talking was selected from each audio file.

The first step in the process is transcribing audio files with high sample frequency. This is done in order to get an accurate transcript from the original audio files. Then, these same audio files are converted to lower sample frequency files. The aim of this research is to find how accurate the transcription of these last types of files is. This is important because it is the key to finding at what frequency audio files are rendered unintelligible, i.e. no words can be extracted or transcribed from the audio. It was decided to set the threshold where no words can be transcribed for privacy reasons. Even if a few words can be picked up by the transcription software, it can't be guaranteed that those specific words do not contain sensitive information. To find the threshold where audio is unintelligible, we will start by down sampling the audio files to 300Hz and then work our way up trying different frequencies and seeing how they affect the transcription of the audio.

3.2.1 Processing high frequency audio

To accomplish successful transcription many steps have to be followed. The most straightforward step was to feed the processed audio into the transcription software. Since the audio was recorded in a social gathering, there was a significant amount of background noise and people speaking in the back. The audios are essentially conversations where the speech of the other people participating in the conversation is also picked up in the recording at a lower volume. This made the transcription software not properly process the speech of the person wearing the microphone. It was clear at this point, that to enhance the audio files before passing them to the transcription software further steps would need to be taken.

One technology that was introduced previously is VAD, used to spot the occurrences of speech and silences in an audio file. In cases where the audio files are long, this technology allows the developer to remove unnecessary silences to make the audio files shorter and easier to work with. In our case, instead of the issue being that the audio files are too long, the main issue is that the audio has too much background noise, making it close to impossible for the transcription software to get an accurate output as previously explained. After trying VAD technology we realized that it was more important for us to improve the audio quality. This method did not improve the outcome from the transcription software so it didn't work in the conditions of this specific research. The alternative step is to find a procedure where the background noise is separated from the foreground noise. To this end, another procedure was considered.

Previously done research by Rafii and Pardo [22] shows encouraging results towards separating background noise from the vocal foreground. By using the Librosa library [23] and following the steps mentioned in its documentation [24], the background noise and vocal foreground were successfully separated and then transcribed. Figure 2 shows the full spectrum of the original audio file at the top, followed by the decomposition of background noise and vocal foreground.

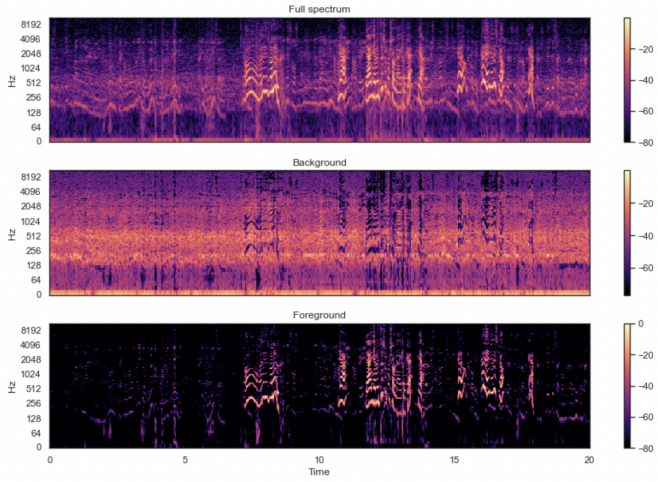

Figure 2: An example of how the vocal foreground is separated from the background noise. From top to bottom, we can see the full audio spectrum, background noise, and the vocal foreground respectively.

In short, the process to remove the background noise grabs the audio files and decomposes them into their corresponding phase and magnitude. Next, it applies filters to separate the background noise and vocal foreground discussed in the librosa documentation [24]. And finally, the spectrum of the vocal foreground noise is converted back to audio and saved locally on the computer.

3.2.2 Processing low frequency audio

The aim of this research is to find and learn how lowering the sampling frequency affects the detection of words, thus affecting the privacy of the speaker. To achieve this goal the following frequencies will be subject to study: 300Hz, 350Hz,

500Hz, 800Hz, 1250Hz, 2000Hz, 3150Hz, 5000Hz, 8000Hz, 12000Hz, 20000Hz, 30000Hz, 44100Hz.

As previously mentioned, the audio files in the dataset are recorded at 44.1kHz. To work with the frequencies listed above, the sampling frequency of audio files needs to be adjusted. To achieve this, we could just use a procedure called down sampling which lowers the sampling frequency of the audio files. This process reduces the sampling frequency by an arbitrary factor M[4]. If the audio that has to be downsampled has frequency components larger than the new frequency, aliasing noise will be introduced in the new audio file. To avoid this phenomenon, it was decided to use a low pass filter instead of directly down sampling the audio files. Low pass filters reduce the bandwidth of the audio replicating the process of down sampling.

3.3 ASR Methods

For the transcription itself, two frameworks are used:

- **Kaldi-NL** An existing model trained for Dutch language.
- **Google Speech Recognition**[5] This is a library offered by Google that supports multiple languages including Dutch.

We decided to use two different pieces of software in order to compare current state-of-the-art technology and see how it performs in Dutch.

To be able to check the results yielded by the ASR technologies a manual transcription was done in the audio files at a sample rate of 44,1kHz.

Transcriptions Using Google Speech Recognition

This software was chosen due to its popularity. Google was one of the first companies to work with speech-to-text technology back in 2005 [25]. Google provides synchronous and asynchronous processes. In this case, we chose to use the synchronous process because it can transcribe audio files that are no longer than 1 minute without having to upload the file to the Google Cloud. Since an End User License Agreement (EULA) was signed in order to work with the dataset, this is the most favorable option in our case. Uploading the audio files to the Google Cloud would violate the clause that refers to further distributing the audio files in the dataset.

After processing the audio following the steps in subsection 3.2, both low and high-frequency audio is fed to the transcription software.

```python
import json
import speech_recognition as SR

audio_file = './path_to_audio'

r = SR.Recognizer()
with SR.AudioFile(audio_file) as source:
    # Set up audio to be transcribed
    audio = r.record(source)
    # Transcribe the audio
    sFinalResult = r.recognize_google(audio, language='nl-NL', show_all = True)
    # Save the response
    response = json.dumps(sFinalResult, ensure_ascii=False).encode('utf8')
    # Transcription text
    transcription = sFinalResult["alternative"][0]["transcript"]
```

Listing 1: Transcribing Audio Using Google Speech Recognition Software

After following the steps in Listing 1 the software returns an output in the following format:

Transcription: 'transcript': 'Nee ik denk dat de natuur dat is natuurlijk helemaal doorgeslagen ja', 'confidence': 0.84188342

Transcriptions Using Kaldi-NL

Since Dutch is not as widely spoken as English, we thought it would be beneficial for the research to look at software specifically targeted at transcribing the Dutch language. We came across a framework developed by the University of Twente and Radboud that gave different models for Dutch [26] using Kaldi as the main transcription software. The model that best accommodated the needs of this research is the one focused on daily conversations due to the nature of the audio files in the dataset.

To set up this software, the steps mentioned in the README.md file on GitHub [26] were followed. This provided an environment where the files were processed using the command line in the computer. This software was run locally and no files were uploaded to the cloud.

3.4 Evaluation

Finally, the results need to be validated. The method used to compare the original transcriptions to the ones generated from the low-frequency audio files is to get their corresponding Word Error Rate. This specific measure is used because it calculates how many "errors" are in the transcription produced by an ASR software compared to the original transcription. The WER of a transcription is calculated with the following formula [27]:

$$WER = \frac{S + D + I}{N}$$

Where S is the number of substitutions, D is the number of deletions, I is the number of insertions and N is the total number of words.

---

[4] For example, if 44.1kHz needs to be turned into 20kHz the factor M would be 2.2 approximately

[5] Note: This is a different library than the Google Cloud Speech API, which requires the user to upload the audio files and would violate the EULA conditions.

4 Experimental Setup and Results

All the steps mentioned in Section 3 have been implemented using Python using Jupyter Notebooks. Another tool used during the process is the command line.

After compiling all the transcriptions from the different sample frequencies we used the WER measurement to compare each transcription to the original transcription. In the beginning, after transcribing the original files at 44.1kHz with Google Speech Recognition the WER was 0.88. This is the main reason we considered a different approach from the one we started with. Now we would like to lay down the different results we got from Kaldi-NL and Google Speech Recognition software. In both cases, to make sure that the software used to get the Word Error Rate is not discriminating against words for being case sensitive, we converted both transcriptions, the original and the one from the transcription software, to lower case.

4.1 Results From Experiments Using Google Speech Recognition Software

In Figure 3 we can see the error each frequency has yielded. For each frequency, all the audios from the 16 different channels are compared to the original transcription and then the errors are averaged out. So for example, to get the WER for 5000Hz the transcriptions for Channel 1 to Channel 16 are compared to the original transcription. Then, the average of the WER of all the channels is computed.

As we can see, the transcription software does a decent job for frequencies higher than 3150Hz. For all these frequencies the WER is still quite high but even then, some information can be extracted from the audio. It might not be much text but there is no way to assure that critical information is not picked up in these words.

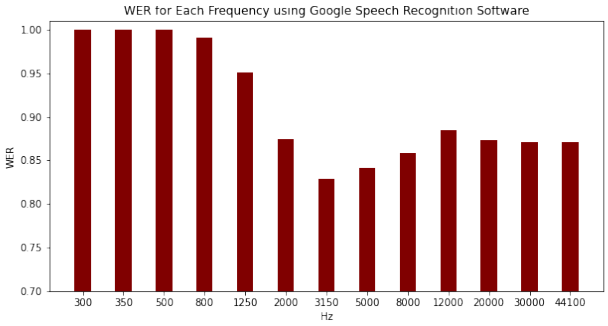

Figure 3: A visual representation of the average WER errors yielded for each frequency using Google Speech Recognition software. The transcriptions from each frequency are compared to the original text of the audios.

4.2 Results From Experiments Using Kaldi Software

Using the same process as with the Google Speech Recognition software and averaging the errors we get the results in Figure 4.

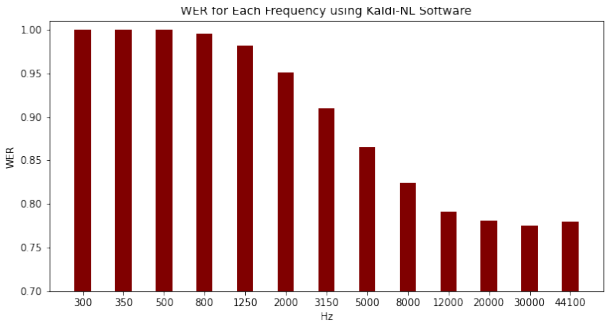

Figure 4: A visual representation of the average WER errors yielded for each frequency using Kaldi software. The transcriptions from each frequency are compared to the original text of the audios.

Finally, in Figure 5, both results have been plotted together to get a clearer view of the performance of each software.

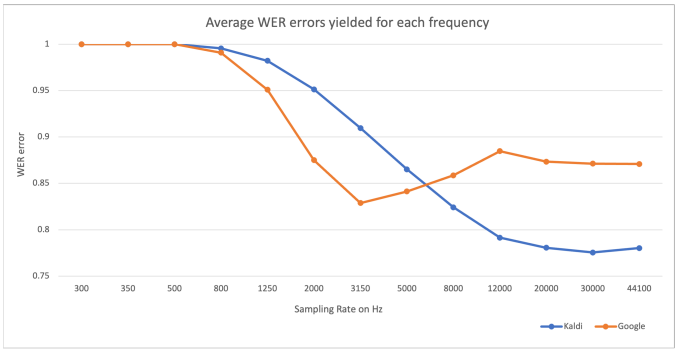

Figure 5: A visual representation of the average WER errors yielded for each frequency using Kaldi and Google Speech Recognition software.

After taking a look at these plots we can see that Kaldi-NL does a better job at transcribing audio in Dutch in the range of 8kHz to 44.1kHz. Overall, Kaldi-NL gets the lowest WER (0,77) at 30kHz. It is interesting to note that Google Speech Recognition software performs better than Kaldi-NL in the range of 800Hz to 5kHz. This could be caused by the fact that Google is targeted at working in the frequencies where speech is found. Speech is primarily located below 4kHz [28] which is the reason why the telephone sample rate is still below 8kHz [29]. One reason why Google's Speech Recognition software might work better below the 8kHz threshold is that the training data might have had audio files retrieved from telephone calls.

On the other hand, Kaldi's performance tends to worsen more linearly. It performs the best at 30kHz and worsens to the point where the transcription is unintelligible (WER of 1) at around 500Hz.

Research done by Kim et al. showed that the average WER error gotten by distant microphones was 0.52 [30]. Distant

microphones refer to when the speaker is talking at a certain distance from the microphone. The lowest WER in this research is 0.77 which could be explained by the language recorded in the audio files and by the conditions of the recording. Additionally, some of the audio files contained words in English. One speaker was American and around 25% of the conversation was in English. When setting up the transcription tool a language is selected, in this case, Dutch. Words that don't correspond to the selected language are not considered. This is a factor that would increase the WER error in the transcriptions.

From this evaluation, we have concluded that in both cases, with Google Speech Recognition Software and Kaldi-NL, the transcription of the audio files is unintelligible at 500Hz. With this sample rate, the WER is 1. This means, that no words could be extracted from the audio files. Automatic Speech Recognition is a tool that it is still developing. It is impossible to assert that in the future if followed these steps, the result would be the same.

5 Responsible Research

This section covers ethical concerns regarding this research. In this case, the main ethical aspects are privacy and reproducibility.

As mentioned before, the EULA terms exclude a free distribution of the materials. This ensures that any personal information or any details mentioned in the conversations by the speakers will not be made public. The privacy of the participants will remain intact. Nevertheless, access can be granted to others who sign the EULA policy as well.

Reproducibility is important since it is what ensures that the scientific community can reproduce and verify the claimed conclusions of this paper. The reproducibility of this research is guaranteed in the frame of the EULA. The one minute long audios together with the original transcriptions will be handed to the research supervisors and will be accessible to anyone that has signed the EULA. Once access to the materials is granted, the methods applied to reach the results of this research are thoroughly explained in section 3 and can be followed by the reader. The results from this work can be fairly compared to any independent study using the same tools.

6 Discussion and Conclusions

The research question in this research was:

How does the reduction in sample frequency hinder the detection of words?

During the process of this research, we have identified the main aspects that could have affected the transcription: the language, the accents, and the environment that the audios were recorded at. These aspects that could not be studied in this research, will be further discussed in section 7 when talking about Future Work.

The EULA signed before getting access to the dataset mentioned that the audio files could not be further distributed without authorization. So in terms of limitations related to the content provided to us, we could not freely distribute the audio to multiple dutch people in order to get an accurate transcription. The only people that could work with the audio were people in the project group and people who had access to the dataset. Also, we were unable to use state-of-the-art technology such as Google Cloud API which accomplishes transcription with only 0.067 WER [31] because by uploading the audio files, we would violate the EULA.

Having in mind that the main researcher does not speak Dutch, there was only one other person in the research team who could provide the original transcriptions of the audio by manually checking the audio files. In the process of manually checking the audio files, there were many cases where some phrases in the audio were unclear, because of different regional accents or overlapping conversations. Having multiple people verify the audios could have resulted in a more accurate transcription via consensus.

Nevertheless, the issue of verifying a transcription from a language that the programmer does not understand brings to front interesting questions about internationalization. Firstly, the difficulty of checking transcriptions in different languages. And secondly, whether the same frequency threshold could be applied to all languages. Complexity in the waveform of vowels and phonemes differs from some languages to others, so the quality of sound could affect some languages more than others in the transcription process.

We can conclude that for both Google Speech Recognizer and Kaldi-NL the audio files were unintelligible at 500Hz and below because the transcription software returned no text. In the future, if the previously mentioned characteristics are taken into account, the threshold of unintelligible audio files could change. So far, it is safe to say that audio recorded at 500Hz and below in Dutch, will provide privacy to the speaker.

7 Future Work

There are still some interesting cases that couldn't be considered thoroughly in this research due to time constraints. For example, what effects do accents have in lower sample frequencies? Early versions of Siri would not recognize audio from people with different English accents, even at high frequencies [32]. Also, would the results obtained in this research be applicable to not widely used languages? Not so widely spoken languages, have fewer data to train models with. Thus, not having models as accurate as of the ones for widely used languages.

There are also people with speech impediments, will these results be accurate in these cases? We haven't worked with audio from people with speech impediments so we were not able to determine how this condition impacts the accuracy of the results obtained in this situation.

As one last point, we have worked with software that needs to have a language predefined before transcribing. In our case, there were English words in the conversations that can not

be transcribed due to the fact that are not native to the selected language. It would be beneficial to learn what results cross language models or automatic language selection software obtains.

Acknowledgement

The author would like to thank Pepijn Vunderink for providing the original transcriptions of the audio files and for providing a working environment with Kaldi-NL.

References

[1] Amazon. *Amazon Alexa Official Site: What is Alexa?* Amazon (Alexa). URL: https://developer.amazon.com/en-GB/alexa.html (visited on June 18, 2022).

[2] "Nuance Communications". *What We Do — Our Mission — Nuance*. Nuance Communications. URL: https://www.nuance.com/company-overview/what-we-do.html (visited on June 7, 2022).

[3] Gus Alexiou. *Voiceitt App For Atypical Speech — A Triumph In Disability Co-Design*. Forbes. Section: Diversity, Equity & Inclusion. URL: https://www.forbes.com/sites/gusalexiou/2021/06/30/voiceitt-app-for-atypical-speech---a-triumph-in-disability-co-design/ (visited on June 7, 2022).

[4] Microsoft. *Dictate in Microsoft 365*. URL: https://support.microsoft.com/en-us/office/dictate-in-microsoft-365-eab203e1-d030-43c1-84ef-999b0b9675fe (visited on June 18, 2022).

[5] Melissa Corrente and Ivy Bourgeault. *Innovation in Transcribing Data: Meet Otter.ai*. 1 Oliver's Yard, 55 City Road, London EC1Y 1SP United Kingdom: SAGE Publications, Ltd., 2022. ISBN: 978-1-5297-9903-3. DOI: 10.4135/9781529799033.

[6] Weijia Xu et al. "A Study of Spoken Audio Processing using Machine Learning for Libraries, Archives and Museums (LAM)". In: *2020 IEEE International Conference on Big Data (Big Data)*. 2020 IEEE International Conference on Big Data (Big Data). December 2020, pp. 1939–1948. DOI: 10.1109/BigData50022.2020.9378438.

[7] L. Canseco, L. Lamel, and J.-L. Gauvain. "A comparative study using manual and automatic transcriptions for diarization". In: *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.* 2005, pp. 415–419. DOI: 10.1109/ASRU.2005.1566507.

[8] Jennifer Yang Hui and Dymples Leong. *The Era of Ubiquitous Listening: Living in a World of Speech-Activated Devices*. SSRN Scholarly Paper 3021623. Rochester, NY: Social Science Research Network, August 18, 2017. URL: https://ssrn.com/abstract=3021623 (visited on June 16, 2022).

[9] Dami Lee. *LaLiga's app listened in on fans to catch bars illegally streaming soccer*. The Verge. June 12, 2019. URL: https://www.theverge.com/2019/6/12/18662968/la-liga-app-illegal-soccer-streaming-fine (visited on June 18, 2022).

[10] Li Wang. "British English-Speaking Speed 2020". In: *Academic Journal of Humanities & Social Sciences* 4.5 (June 16, 2021). Publisher: Francis Academic Press. DOI: 10.25236/AJHSS.2021.040517.

[11] Clive Thompson. "AI, the Transcription Economy, and the Future of Work". In: *Wired* (). Section: tags. ISSN: 1059-1028. URL: https://www.wired.com/story/ai-transcription-economy-future-of-work/ (visited on April 19, 2022).

[12] ConfLab Team. "The Socially Perceptive Computing Lab". MA thesis. 2019. URL: https://conflab.ewi.tudelft.nl.

[13] Henning Schulzrinne. Explanation of 44.1 kHz CD sampling rate. January 10, 2008. URL: https://www1.cs.columbia.edu/~hgs/audio/44.1.html (visited on June 19, 2022).

[14] Microphone University. *Facts about speech intelligibility: human voice frequency range*. DPA. March 13, 2021. URL: https://www.dpamicrophones.com/mic-university/facts-about-speech-intelligibility (visited on June 16, 2022).

[15] Mark Gales and Steve Young. "The Application of Hidden Markov Models in Speech Recognition". In: *Foundations and Trends® in Signal Processing* 1.3 (February 20, 2008). Publisher: Now Publishers, Inc., pp. 195–304. ISSN: 1932-8346, 1932-8354. DOI: 10.1561/2000000004.

[16] Kan Li and Jose Principe. "Biologically-Inspired Spike-Based Automatic Speech Recognition of Isolated Digits Over a Reproducing Kernel Hilbert Space". In: *Frontiers in Neuroscience* 12 (March 2018). DOI: 10.3389/fnins.2018.00194.

[17] Gamal Bohouta and Veton Këpuska. "Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx)". In: *Int. Journal of Engineering Research and Application* 2248-9622 (March 1, 2017), pp. 20–24. DOI: 10.9790/9622-0703022024.

[18] Delano de Ruiter. "Sentiment Analysis on Dutch Phone Call Conversations". MA thesis. The Netherlands: University of Amsterdam, 2019.

[19] Radboud University. *PhD defence: Automatic Speech Recognition in noisy environments and with heavy accents*. Centre for Language Studies. Last Modified: 2021-09-09. URL: https://www.ru.nl/cls/news-events/news/news/phd-defence-asr-with-noise-and-accent/ (visited on June 18, 2022).

[20] Zheng-Hua Tan, Achintya kr. Sarkar, and Najim Dehak. "rVAD: An unsupervised segment-based robust voice activity detection method". In: *Computer Speech & Language* 59 (January 1, 2020), pp. 1–21. ISSN: 0885-2308. DOI: 10.1016/j.csl.2019.06.005.

[21] Oren Lederman. *Rhythm Badge*. MIT Media Lab. 2018. URL: https://www.media.mit.edu/posts/rhythm-badge/ (visited on June 18, 2022).

[22] Zafar Rafii and Bryan Pardo. "Music/Voice Separation Using The Similarity Matrix". In: (2012), p. 6.

[23] Brian McFee et al. “librosa: Audio and music signal analysis in python”. In: *Proceedings of the 14th python in science conference*. Vol. 8. 2015.

[24] Librosa Development Team. *Vocal separation — librosa-gallery 0.1.0 documentation*. URL: https://librosa.org/librosa_gallery/auto_examples/plot_vocal_separation.html (visited on May 7, 2022).

[25] Google. *How one team turned the dream of speech recognition into a reality - Google Careers*. URL: https://careers.google.com/stories/how-one-team-turned-the-dream-of-speech-recognition-into-a-reality/ (visited on June 7, 2022).

[26] Emre Yilmaz Maarten van Gompel. *Automatic Speech Recognition for Dutch*. DOI: 10.5281/zenodo.6621815.

[27] Martin Thoma. *Word Error Rate Calculation*. November 15, 2013. URL: https://martin-thoma.com/word-error-rate-calculation/ (visited on June 19, 2022).

[28] Pery Pearson. *Sound Sampling*. URL: http://www.hitl.washington.edu/projects/knowledge_base/virtual-worlds/EVE/I.B.3.a.SoundSampling.html (visited on June 18, 2022).

[29] Sound Design. *Sample Rate & Audio Sampling – Explore Each Concept & 4 Types*. Become better creators — together. May 5, 2022. URL: https://academy.wedio.com/sample-rate/ (visited on June 18, 2022).

[30] Joshua Y. Kim et al. *A Comparison of Online Automatic Speech Recognition Systems and the Nonverbal Responses to Unintelligible Speech*. arXiv:1904.12403. type: article. arXiv, April 28, 2019. arXiv: 1904.12403[cs,eess]. URL: http://arxiv.org/abs/1904.12403 (visited on June 18, 2022).

[31] Chung-Cheng Chiu et al. *State-of-the-art Speech Recognition With Sequence-to-Sequence Models*. arXiv:1712.01769. type: article. arXiv, February 23, 2018. DOI: 10.48550/arXiv.1712.01769. arXiv: 1712.01769[cs,eess,stat].

[32] Allison Koenecke et al. “Racial disparities in automated speech recognition”. In: *Proceedings of the National Academy of Sciences* 117.14 (April 7, 2020). Publisher: Proceedings of the National Academy of Sciences, pp. 7684–7689. DOI: 10.1073/pnas.1915768117.