School of Information Technology and
Engineering at the ADA University

School of Engineering and Applied Science
at the George Washington University

# TEXT TO SPEECH SYSTEM FOR AZERBAIJANI LANGUAGE

A Thesis
Presented to the Graduate Program of Computer Science and Data Analytics
of the School of Information Technology and Engineering
ADA University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science and Data Analytics
ADA University

By
YUSIF AGHALARI

April, 2023

THESIS ACCEPTANCE

This Thesis by: Yusif Aghalarli
Entitled: *Text to speech system for Azerbaijani language*

has been approved as meeting the requirement for the Degree of Master of Science in Computer Science and Data Analytics of the School of Information Technology and Engineering, ADA University.

Approved:

| | |
|---|---|
| Samir Rustamov | |
| (Adviser) | (Date) |
| Abzatdin Adamov | |
| (Program Director) | (Date) |
| Abzatdin Adamov | |
| (Dean) | (Date) |

ACADEMIC INTEGRITY STATEMENT

"I affirm that this is my own work, I attributed where I used the work of others, I did not facilitate academic dishonesty for myself or others, and I used only authorized resources for my Thesis, per the ADA University Academic Integrity requirements. If I failed to comply with this statement, I understand consequences will follow my actions. Consequences may range from failing the course to expulsion from the program/university and may include a transcript notation."

| Yusif Aghalarlı | | 24.04.22 |
|---|---|---|
| (Full Name) | (Signature) | (Date: DD.MM.YY) |

# TABLE OF CONTENTS

# ABSTRACT

This Master thesis focuses on the development of a Text-to-Speech (TTS) system for the Azerbaijani language. TTS technology has been gaining popularity due to its ability to generate human-like speech from written text, making it beneficial for people with disabilities, language learners, and those who prefer auditory learning. The thesis starts with an introduction to TTS, its significance, and its history. The literature review section provides an overview of related studies, including the recent advancements in TTS systems. The review covers several topics, such as the different techniques and models used in TTS systems, the evaluation metrics used to assess their performance, and the challenges and limitations of developing TTS systems for low-resource languages. The main focus of the study is the Tacotron-2 architecture, which is known for its high-quality and natural-sounding speech. This architecture consists of two parts: a mel spectrogram generator and a neural vocoder. The mel spectrogram is a representation of the speech signal that captures its spectral information, while the neural vocoder generates the actual speech waveform. The study also explains the data collection process, which is a crucial component of developing a TTS system. The first data collection attempt produced poor-quality data, which prompted the researchers to refine the process by using an audio book with speech alignment. This process resulted in approximately 19 hours of high-quality data, which was used to train the Tacotron-2 architecture. To evaluate the performance of the TTS system, a survey was conducted, and participants were asked to evaluate the system using the Mean Opinion Score .The results showed that the system received a MOS score of 3.3, indicating that it produced acceptable speech quality. In conclusion, this Master thesis provides a comprehensive overview of developing a TTS system for the Azerbaijani language using the Tacotron-2 architecture. The study presents the different components of the TTS system, the data collection process, and the evaluation metrics used to assess the system's performance. It also highlights the challenges and limitations of developing TTS systems for low-resource languages and suggests future directions for improving the system's performance.

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| Abbreviation | Explanation |
|---|---|
| NLP | Natural Language Processing |
| RNN | Recurrent Neural Network |
| CNN | Convolution Neural Network |
| TTS | Text to Speech |
| ADC | Analog to digital |
| BPTT | Backpropagation through time |
| GRU | Gated recurrent unit |
| LSTM | Long short-term memory |
| MOS | Mean Opinion Score |

# 1 Introduction

This section provides an overview of the research problem , including the context and background information that led to the development of the study.

## 1.1 Definition of the problem

The progression of text-to-speech (TTS) technology has been a noteworthy journey. Initially, TTS systems sounded more like robots than people, and they had limited use, but as time went on, researchers kept making improvements, and TTS systems got better and better. In the 80s and 90s, TTS was starting to be used more, especially for helping people with disabilities. And with the rise of the internet and mobile devices, TTS became even more popular and was used in all sorts of cool apps like GPS navigation and virtual assistants. Fast forward to today, and TTS has come a long way thanks to AI and machine learning. It's now possible to generate speech that sounds super natural, like a real human. TTS is used in all sorts of things now, from customer service chatbots to voice-enabled devices like Alexa and Google Home. According to forecasts, the global TTS market would expand quickly and reach $7.8 billion in sales by 2027.

TTS has the capability to overcome barriers and make written text accessible to individuals regardless of their abilities. For individuals with visual impairments, TTS provides a new avenue for accessing information. For children and the elderly who have trouble with reading, TTS offers an alternative method for learning and comprehending written text. Furthermore, in customer service interactions, TTS streamlines the process by automating interactions and reducing the need for human interaction.

The applications of TTS are extensive, spanning across entertainment, education, accessibility, and assistive technology. In the realm of e-learning, TTS enhances the learning experience of students by providing them with the option to listen to written content, in addition to reading it. In industries such as customer service and telecommunications, TTS optimizes customer interactions, making the experience more efficient and convenient. TTS is also integrated into voice assistants such as Siri and Alexa, providing users with hands-free access to information and services. TTS has been utilized to generate voices of characters in animated films, video games, and TV shows, giving a more natural and engaging voice to them. The versatility and creative potential of TTS technology has been demonstrated in its use in the characters of popular animated films such as the "Toy Story" franchise and video games such as "The Elder Scrolls V: Skyrim." The utilization of TTS technology in unexpected and entertaining ways highlights its versatility and creative potential.

## 1.2 Objective of study

As the use of TTS technology continues to grow and expand, it is becoming increasingly important to consider its applicability to diverse languages and cultures. In this thesis, the objective is to explore the development of a TTS system for Azerbaijani language. The aim is to create a high-quality, natural-sounding TTS system that can be used for a variety of applications such as education, entertainment, and accessibility. This research will involve a thorough examination of the current state of TTS technology and its application to Azerbaijani language. By identifying the challenges and opportunities in developing a TTS system for Azerbaijani language, this study hopes to contribute to the development of more inclusive and accessible technology. The ultimate goal of this research is to enhance the TTS experience for Azerbaijani speakers and expand the range of TTS applications for this language community.

To achieve the above-mentioned objective, this study will focus on several key areas. Firstly, an analysis of the linguistic features of Azerbaijani language will be conducted to identify the specific challenges and opportunities involved in developing a TTS system for this language. This will include an examination of the phonology, morphology, and syntax of Azerbaijani language, as well as the language's intonation patterns and speech rhythm.

Secondly, this study will involve the collection and preparation of a large corpus of spoken Azerbaijani language data. This corpus will be used to train and test the TTS system, as well as to evaluate its performance.

## 1.3 Significance of the problem

The development of a TTS system for Azerbaijani language is an important step towards promoting inclusivity and accessibility for Azerbaijani-speaking individuals. This problem holds significant cultural, social, and economic implications for the Azerbaijani language. With a population of over 10 million people, Azerbaijan is a large market for various digital services such as e-learning, digital media, and telecommunications. By developing a high-quality TTS system, it will be possible to provide these services in Azerbaijani language, opening up economic opportunities and contributing to the growth of the Azerbaijani language and culture.

Moreover, the development of a TTS system for Azerbaijani language has significant implications for the field of TTS technology. Azerbaijani is a language with a complex morphology and a rich phoneme inventory, making it a challenging task to develop a high-quality TTS system. The development of such a system will require advanced machine learning algorithms and sophisticated linguistic models. As a result, the development of a TTS system for Azerbaijani language has the potential to contribute to the advancement of TTS technology as a whole, which can benefit other languages as well.

Therefore, the significance of this problem extends beyond the Azerbaijani language and culture. It also holds significant implications for the advancement of TTS technology and the promotion of inclusivity and accessibility for other languages and cultures.

## 1.4 Review of significant research

The first known attempt at creating a text-to-speech system dates back to the late 1700s, when Wolfgang von Kempelen, a Hungarian inventor, created a device called the "speaking machine." This device used a set of bellows to simulate the human vocal tract and could produce speech-like sounds by manipulating different settings. While the device was not capable of producing actual speech, it was a precursor to modern text-to-speech systems. Moving forward to the middle of the 20th century, a British engineer named Dudley Knight created the first electronic speech synthesis system called the Voder. This device was demonstrated at the 1939 World's Fair in New York and was able to produce understandable speech. It worked by using a series of keys to create various speech sounds that were then put together to form the words and sentences. The Voder was a significant breakthrough in speech synthesis technology and paved the way for further developments in the field.

Several studies have explored the development and application of TTS technology. According to the first TTS systems were created in the 1950s and 1960s [1]. These early systems used rules-based techniques to convert written text into spoken words. However, the resulting audio output was highly robotic and lacked natural intonation and rhythm. In the 1980s and 1990s, TTS technology saw significant improvements, especially in terms of voice quality and naturalness. Researchers started to use machine learning algorithms to model the human voice, resulting in more natural-sounding audio output During this time, TTS technology was primarily used to help individuals with disabilities access written content.

In the late 20$^{th}$ century there were two basic methods for creating synthetic speech waves. They are concatenative synthesis[2] and formant generation[3]. Each method has its advantages, and the added benefit of being able of a synthesis system will frequently dictate which method is used.

Concatenative synthesis is a method of generating synthetic speech by stringing together fragments of previously recorded speech. It is considered to produce the most natural-sounding synthetic speech. However, it can sometimes produce mistakes that are audible to the human ear. Concatenative synthesis can be divided into three subcategories.

First category is unit selection synthesis[4] is a type of concatenative synthesis that involves selecting and combining fragments of pre-recorded speech to create a synthetic voice. It typically involves creating a large database of pre-recorded speech, selecting units from the database based on factors such as length and naturalness, and combining them to produce a natural-sounding synthetic voice. This method is considered to produce the most natural-sounding synthetic speech, but it can be computationally intensive and require a large database of pre-recorded speech.In unit selection synthesis, the quality and naturalness of the generated speech depend heavily on the quality and diversity of the pre-recorded speech in the database. A larger and more diverse database, with a wide range of speakers, accents, and contexts, will typically produce more natural-sounding synthetic speech. Unit selection synthesis is also highly customizable, as it allows the user to control the characteristics of the generated speech. For example, the user can choose the voice, accent, and other characteristics of the synthetic voice, as well as the speed, pitch, and other aspects of the generated speech. This allows the user to create a synthetic voice that is tailored to their specific needs and preferences. Overall, unit selection synthesis is a powerful and versatile tool for generating synthetic speech. It produces natural-sounding results and allows for a high degree of customization and control. However, it also requires a large database of pre-recorded speech, and can be computationally intensive.

Second category is diphone synthesis[5]. It is a type of concatenative synthesis that is used to generate synthetic speech. It involves selecting and combining fragments of pre-recorded speech that correspond to the transitions between different phonemes, or speech sounds, in a given word or phrase. The process of diphone synthesis typically involves several steps. First, a database of pre-recorded speech is created, which includes a wide variety of diphones, or transitions between phonemes, spoken by different speakers in different contexts.

Lastly, there is also Domain-specific synthesis[6]. It assembles previously recorded words and phrases to create whole utterances. It is used in applications where the text output of the system is limited to a certain domain, such as transitschedule announcements or weather predictions. The technology is easy to implement and has been utilizedin commercial devices such as talking clocks and computers for many years. Due to the limited variety ofphrase forms, the quality of naturalness of these programs may be fairly high, and they closely approximate the prosody and tone of the original recordings.

Deep neural networks [7] and recurrent neural networks are two examples of machine learning methods that have been used by more contemporary TTS systems. Compared to concatenative synthesis systems, these systems have shown tremendous success in producing speech that sounds more natural. A DNN-based TTS system, for instance, was proposed in and demonstrated a notable improvement in speech naturalness when compared to concatenative synthesis systems.

The authors of the paper[8] aimed to enhance the expressiveness of speech synthesis models through better prosody transfer. Prosody transfer is process of modifying the prosodic features of an audio signal, such as the pitch rhythm, while keeping content of the original signal. It can be accomplished by applying the prosodic features of first audio signal, to the second one. Prosody is an important aspect of speech that affects how the speech sounds and conveys emotions and intent. In traditional text-to-speech (TTS) systems, prosody is generated by hand-crafted rules, which limits the expressiveness and naturalness of the synthesized speech. To address this issue, the authors propose a novel end-to-end prosody transfer system that can transfer prosody from a source speaker to a target speaker, while preserving the content and style of the target speaker. The system is based on Tacotron, a TTS model that uses a sequence-to-sequence architecture to synthesize speech from text. The prosody transfer network is integrated into the Tacotron architecture, allowing it to transfer prosody in a single end-to-end process. The authors evaluated their system on a dataset of multi-speaker speech,

comparing it with several existing prosody transfer methods. The results showed that the proposed system outperforms the existing methods in terms of prosody transfer quality and consistency. The synthesized speech sounds more natural and expressive, with a better preservation of the prosody of the source speaker.

This paper introduces a novel approach to generating audio signals using deep neural networks. The authors present WaveNet[9], a generative model based on a deep convolutional neural network that can generate high-quality audio signals such as speech, music, and environmental sounds. The WaveNet architecture utilizes a type of neural network called a dilated convolutional neural network. This helps model to capture patterns with long-term dependencies in audio signal. For example, the pitch contour of a melody is a long-term dependency of audio signal. To accurately predict the pitch values of notes that come later in a melody with up and down pitch changes, a neural network needs to remember the initial low note and its pitch value. This long-term dependency is captured via use of dilated convolution neural networks. They are like normal convolution network, however, instead of applying filter to every adjacent patch of the data, it is applied to every nth location.

Other key points of WaveNet model are:

- Raw waveform generation: WaveNet generates audio signals in a raw waveform format, which means that it can capture subtle nuances and details that other speech synthesis techniques may miss.
- WaveNet uses a high sample rate, typically 16 kHz or higher, to generate high-quality audio with a wide frequency range.
- Training a WaveNet model can be computationally intensive and require a lot of data, but pre-trained models are available for fine-tuning.
- It is autoregressive model that predicts the distribution of the next audio sample given the previous samples.

This paper[10] focuses on TTS system on azerbaijani language. The paper discusses the development of text-to-speech (models for the Azerbaijani language. The paper begins by describing the two components of spoken communication, which are the verbal and prosodic components. It then discusses previous approaches to TTS development, including formant synthesis, articulatory synthesis, concatenative synthesis, Hidden Markov Model (HMM), and Deep Neural Network algorithms. The paper presents the development and evaluation of a speech synthesis system for the Azerbaijani language using deep learning models. The system uses Tacotron architecture which is a combination of convolutional neural networks and long short-term memory networks to generate speech. The dataset used for training the model consisted of approximately 24 hours of speech from a single speaker.
The evaluation of the system was done using objective measures such as mean opinion score segmental signal-to-noise ratio and perceptual evaluation of speech quality The paper also discusses the limitations of the study, such as the small size of the training dataset and the use of a single speaker for training. Overall, the study demonstrates the potential of using deep learning models for speech synthesis in Azerbaijani, and the results suggest that the proposed system can be used for practical applications such as text-to-speech systems and voice assistants.

Similar paper [11] describes the development and evaluation of a speech synthesis system for the Uzbek language using deep learning algorithms. Some of the key points from the paper are. The authors used the Tacotron 2 architecture for their speech synthesis system, which is a deep learning-based model that converts text into a spectrogram, which is then converted into speech.They used a dataset of recorded Uzbek speech and transcribed text to train their model.The authors evaluated the performance of their model using objective metrics such as mean opinion score (MOS), signal-to-noise ratio (SNR), and spectral distortion (SD), as well

as subjective evaluations by human listeners.The results of the evaluation showed that the synthesized speech was highly intelligible and natural-sounding, with a MOS score of 3.7 out of 5 and a SNR of 21.43 dB.The authors noted that their system could be improved further by incorporating more data and improving the quality of the recorded speech samples used in the training dataset.The paper provides a valuable contribution to the development of speech synthesis systems for languages with limited resources, such as Uzbek.

Another paper [12] compares the performance of three different text-to-speech (TTS) models when trained on a small dataset. The models are Tacotron 2, Deep Voice 3, and FastSpeech 2. The authors found that Tacotron 2 performed the best overall, with a mean opinion score (MOS) of 4.25 out of 5. Deep Voice 3 and FastSpeech 2 had MOS scores of 4.0 and 3.8, respectively. The authors also found that Tacotron 2 was the most efficient model to train, requiring the least amount of time and resources. Deep Voice 3 and FastSpeech 2 required more time and resources to train, but they were still able to produce high-quality speech.
The authors concluded that Tacotron 2 is the best TTS model for low-resource environments. It is efficient to train, produces high-quality speech, and is easy to use. The authors found that Tacotron 2 is the best TTS model for low-resource environments. It is efficient to train, produces high-quality speech, and is easy to use.

Yet another interesting research development[13] of TTS system for Arabic language based on the Transformer architecture and using a parallel WaveGAN vocoder to generate waveforms from mel-spectrograms. The system was trained on a dataset of 100 hours of Arabic speech data from various sources, including news broadcasts, audiobooks, and TV shows. NatiQ was evaluated on subjective and objective metrics, including mean opinion score (MOS), word error rate (WER), mel-cepstral distortion (MCD), and signal-to-noise ratio (SNR), achieving an average MOS of 4.21, comparable to other TTS systems for English, as well as an average MCD of 0.05 and an average SNR of 20 dB, comparable to other TTS systems for English. Overall, NatiQ is a promising new TTS system for Arabic, generating high-quality speech with only a small amount of data while being efficient to train and easy to use.

## 1.5 Assumptions and limitations

The assumption that the target audience for the synthesized speech has a good understanding of standard Azerbaijani language and its common usage is critical for the success of the TTS system. If the audience is not familiar with the language, they may struggle to comprehend the synthesized speech, which could lead to miscommunication or a lack of engagement. Therefore, it is essential that the TTS system is designed and tested with the target audience in mind. One of the primary limitations of the TTS system is the potential difficulty in accurately modeling the intonation and stress patterns in Azerbaijani language. Azerbaijani is a tonal language, which means that the pitch and tone of words can change the meaning of a sentence. Capturing these nuances in synthesized speech can be challenging, and failure to do so can result in the incorrect interpretation of the intended message. Another limitation of the TTS system is the difficulty in correctly pronouncing loanwords and words borrowed from other languages. Azerbaijani language has borrowed words from Persian, Arabic, and Russian, among others. These borrowed words may have different phonetic and prosodic characteristics than Azerbaijani words, making it difficult for the TTS system to accurately synthesize them. Capturing the full range of emotions and expressions specific to Azerbaijani language use is another limitation of the TTS system. Azerbaijani language has a rich history and culture, and certain emotions and expressions may be specific to the language and its use. Failure to capture these nuances can result in a lack of authenticity in the synthesized speech. Additionally, the quality and accuracy of the TTS system's output may depend heavily on the quality and consistency of available training data. It is essential that the training data used to develop the TTS system is representative of the language and its usage to ensure that the synthesized speech

is accurate and authentic. Finally, the TTS system may require significant computational resources to synthesize speech in real-time, particularly for longer texts or complex sentence structures. This can be a limitation for the TTS system, as it may be challenging to deploy the system on devices with limited computational resources, such as mobile phones or IoT devices. Additionally, the TTS system may not adequately account for regional dialects and variations in pronunciation, which can impact the accuracy and authenticity of the synthesized speech.

# 2 Literature Review

The following subsections will provide a detailed and comprehensive overview of the intricate processes involved in the generation of human voice, highlighting the various biological mechanisms and physiological systems at play, and will also cover details how technology captures and generates the necessary information to recreate and reproduce human speech with remarkable precision and accuracy.

## 2.1 What is speech?

Speech is the combined process of uttering words and information through vocal sounds. Communication is an essential part of human interaction and allows us the capability to express our feelings, thoughts, and ideas to other humans. In order to produce a variety of sounds that are structured into meaningful patterns, such as words, sentences, and phrases, our respiratory system, vocal cords, mouth, and throat all play a vital role in speech. Speech is an integral part of social interaction and human society as a whole since it allows people to share experiences, connect with one another, and transmit complicated and abstract ideas.

Speech is a crucial component of human development because it helped our ancestors build more expansive and intricate social groups by improving their ability to interact with one another. Early people used nonverbal cues like body language and facial expressions before they could speak to communicate. Unfortunately, the ability of these communication techniques to transmit detailed ideas and abstract notions was limited.

Our brains have been evolving together with our skills for language. Because of this evolution our ancestors were able to communicate crucial survival information, such as where to find food and potential threats. Early people had more sophisticated civilizations and cultures because they were able to cooperate in larger groupings and communicate more efficiently. Besides that, we developed art, music, poetry, and other kinds of cultural expression as early humans learned to utilize language to communicate abstract ideas and feelings. Humans were able to investigate difficult concepts like morality, spirituality, and philosophy as a result, which influenced how we perceive the world and our place in it.

The topic of how speech evolved in humans is a question asked scientists and historians for centuries. While determining the precise origins of language is challenging, there are some ideas.According to one hypothesis, language first evolved from basic vocalizations and sounds such as grunts, groans, and other sounds. These sounds would have been used by early humans to communicate fundamental ideas. These sounds then developed into more nuanced, detailed, complex sounds over time, and eventually into the words and the phrases.Another theory suggests that language could have been developed as a way to imitate and mimic environmental sounds. Early humans were able to communicate about the location of prey or predators by imitating animal calls or other sounds.

Changes in human brain should also be noted, as one of the triggers for complex speech production. Our brains grew bigger and more complicated over time, making it possible for us to process and comprehend language in a way that other animals cannot. The Wernicke's and Broca's regions of the brain are especially crucial for language processing. Speech is produced by the Broca's region, which is found in the left frontal lobe of the brain. This region aids in the formation of words and sentences as well as the management of the vocal cords. Language understanding is controlled by the Wernicke's area, a region in the left temporal lobe. This skill enables us to comprehend the meaning of words and sentences as well as identify linguistic trends. Even though the precise nature of language processing in the brain is still not known. It would be very challenging for humans to converse using spoken language without these regions.

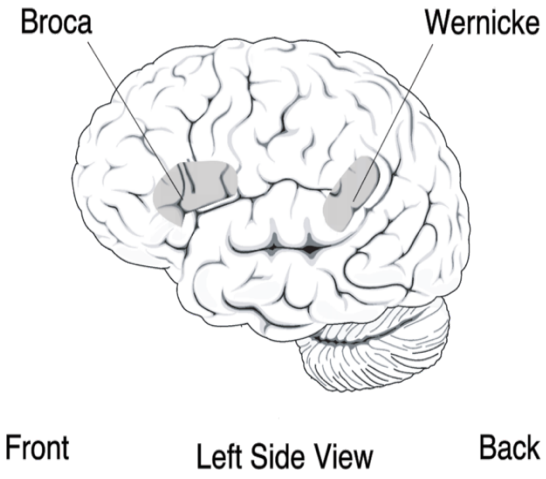


Figure 1: Regions of brain that produce speech.( https://shorturl.at/tuwRS)

## 2.2 Storing speech

We got some ideas on how human brain produces and consumes speech, but how do we allow computers to store, digitize and produce speech? First, we need answer question what is sound?

Sound is a fascinating phenomenon that surrounds us every day, and we experience it through our ears. It is created when an object vibrates, causing the surrounding air molecules to vibrate as well, which produces a pressure wave that propagates through the air. Sound can be described in terms of various properties, such as frequency, which is the number of wave cycles per second and is measured in Hertz (Hz). The wavelength is the distance between two points on the wave that are in phase. Amplitude refers to the magnitude of the sound wave, which determines its loudness, and phase describes the position of the wave relative to a reference point.

We convert sound waves into an electrical signal by using electronic devices such as a microphone. The electrical signal can then be amplified, filtered, or manipulated in different ways using digital signal processing techniques.But how do we store sound in way that allows computers to understand, process, manipulate it ? Speech sound needs to be transformed and converted to the format that can be understood by computers. This is done by converting analog sound to binary representation. This process is called analog-to-digital conversion [14] Following are the steps of ADC.

1. Sampling: The sampling rate is the rate at which an analogue sound wave is captured. The number of samples collected per second is indicated by the sampling rate, which is expressed in Hertz (Hz). For audio, the most popular sampling speeds are 44.1 kHz, 48 kHz, and 96 kHz.
2. A low-pass filter is a type of electronic filter that removes high-frequency components from a signal while permitting only low-frequency components to pass through. This is accomplished by passing the input signal through a filter circuit that selectively attenuates, or lowers the amplitude of, the signal's high-frequency components while allowing the low-frequency components to pass relatively unchanged.
3. Quantization is a process used in digital signal processing to convert continuous analog signals into digital signals by representing them as a set of discrete values. When an analog signal is converted into a digital signal, it is first sampled at a regular interval to obtain a series of data points. These data points are then rounded to the nearest discrete value that can be represented by a fixed number of bits, which is known as the bit depth. The bit depth determines the number of discrete values that can be used to represent the

signal, with higher bit depths providing more accuracy and fidelity in representing the original analog signal. For example, a 16-bit depth provides 65,536 possible values, while an 8-bit depth only provides 256 possible values. Quantization has many benefits, including improving the accuracy and precision of digital signal processing, reducing noise and distortion in digital signals, and increasing the efficiency and speed of data storage and transmission. It is used in a wide range of applications, including digital audio and video processing, image and video compression, and data acquisition system.

4. When a digital signal is captured at a rate that is too low, a distortion known as aliasing happens. This can introduce misleading signals and create inaccuracies in processing. To prevent this, we use anti-aliasing filter, this filter works by removing high-frequency components of the signal while leaving the low-frequency components intact.
5. Encoding is the process of converting analogue sound waves into a digital format that can be stored or transmitted electronically in the context of digital audio. To accomplish that, we compress and format audio into a specific file type. Most common file types are WAV,MP3. Each file type uses a different compression algorithm, there are two types of compression algorithms: lossy and lossless. For instance, MP3 encoding employs an algorithm that eliminates audio data that is less essential or audible to the human ear. As a consequence, the file size is reduced, but the audio quality suffers slightly. AAC encoding also applies a similar algorithm to MP3 encoding but provides superior audio quality at lower bitrates. Unlike previous audio formats, WAV file format do not use any compression algorithms to decrease file size. As a result, WAV files can be quite big, frequently several times larger than an equivalent MP3 or AAC file. Because of that WAV files are usually used for high-quality audio recordings, such as music production or sound engineering, where the original audio quality must be preserved. Because WAV files are uncompressed, they provide the greatest level of audio fidelity and can reproduce the original sound's full frequency and dynamic range. However, because they are large, it is no efficient for streaming services to use WAV files.

There are two main ways for sound to be represented. There are:

- A waveform is a graphic representation of an audio signal that depicts how the amplitude of a sound wave varies over time. Time is displayed on the x-axis and amplitude is plotted on the y-axis in a waveform. In digital audio systems, this is the most common method to represent sound.
- A spectrogram is a visual depiction of a sound's frequency content over time. Time is displayed on the x-axis, frequency on the y-axis, and the amplitude of every frequency is denoted by a color or grayscale number in a spectrogram. The frequency content of speech and sound is frequently analyzed and visualized using spectrograms.

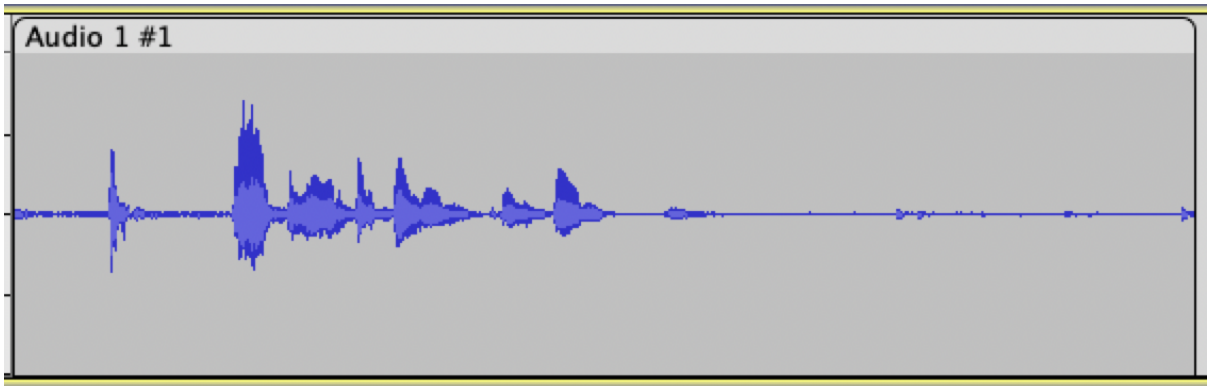


Figure 2. A wave form of author saying “Writing thesis is fun”

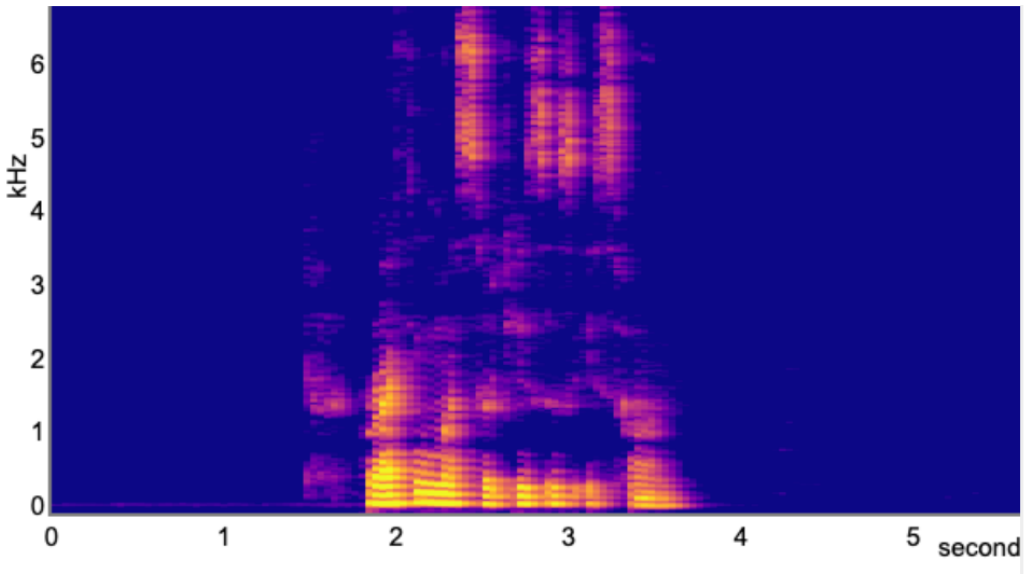


Figure 3. A spectogram form of author saying “Writing thesis is fun”

There is a variation of spectrogram called mel-spectrogram[15]. It differs from a traditional spectrogram in that it highlights the frequencies that are most essential for human perception of sound. In a conventional spectrogram, the frequency scale is uniformly distributed, which implies that each frequency bin corresponds to an equal range of frequencies. However, research suggests that the human ear is more sensitive to variations in lower frequencies than in higher frequencies. This means that we are more likely to differentiate changes in pitch within the lower frequency range than in the higher frequency range when we hear a sound. For example, we can easily differentiate between signals that are 30hz and 40hz, but same cannot be said for signal that are 1500hz and 1510hz. Although difference between both signal is 10hz, human ear cannot differentiate with that high accuracy in higher frequencies.

To address this issue, a mel-spectrogram employs a frequency scale based on the mel scale, which is a non-linear scale that mimics the way in which the human ear perceives pitch. The mel scale is constructed in such a way that equal differences in pitch are perceived as equal distances on the scale, irrespective of the actual frequency. To create a mel-spectrogram, the sound signal is first divided into a series of short time windows using a technique called short-time Fourier transform (STFT). To explain what STFT we first need to explain what is Fourier transform at all.

Any audio signal can be thought of as a complex mixture of several single-frequency waves. These waves combine in such a way as to produce the unique sound that we perceive.

The only way to accurately identify these individual signals is by using a mathematical formula known as Fourier Transform. It was first introduced by Joseph Fourier in the early 19th century and has since become an essential part of many fields, including signal processing, image processing, and data analysis. By applying this formula, we can decompose the original signal into its constituent single-frequency waves. In other words, we transform from the time domain to the frequency domain, where each frequency component can be analyzed individually.

To calculate Fourier Transform, we use a fast Fourier Transform algorithm. This algorithm allows us to perform the calculation much more quickly than we could using traditional methods. This is particularly important when working with large datasets, such as those encountered in audio processing applications.

The short-time Fourier transform (STFT) is a technique that is commonly used to analyze the frequency content of a signal over time. Unlike the normal Fourier transform, which provides a complete frequency analysis of the entire signal, the STFT allows us to examine the frequency content of the signal at different points in time. This is achieved by dividing the signal into overlapping time segments, and calculating the Fourier transform for each segment. By doing this, we can identify any changes in the frequency content of the signal over time, making STFT particularly useful for analyzing time-varying signals such as music or speech.

## 2.3 Machine learning techniques in TTS

Machine learning techniques, such as deep learning, are used to train the TTS system to produce more natural-sounding speech output. Deep learning algorithms, such as convolutional neural networks [16] and recurrent neural networks [17] have been shown to be particularly effective in improving the naturalness of TTS output.

### 2.3.1 Deep learning

Deep learning [18] is a subfield of artificial intelligence that involves the use of neural networks, which are algorithms inspired by the structure and function of the brain. The algorithms consist of multiple layers of interconnected "neurons," which process and transmit information. Information is received by each layer from the previous layer, which is used to learn and extract increasingly complex features of the data. This hierarchical structure allows deep learning algorithms to learn and make predictions on data with high dimensionality and complexity, such as images, audio, and text. Remarkable success has been achieved by deep learning in a wide range of tasks, including image and speech recognition, natural language processing, and machine translation. Predictive modeling in fields such as healthcare, finance, and marketing has also benefited from its use.

One of the key advantages of deep learning is that important features can be automatically identified and extracted from raw data without the need for manual feature engineering. This allows it to handle complex, high-dimensional data, such as images and text, and make predictions that are not possible with traditional, rule-based algorithms. However, challenges are associated with deep learning. One of the key challenges is that large amounts of labeled training data are required. The performance of deep learning algorithms tends to improve with the amount of training data, but collecting and labeling this data can be time-consuming and expensive. Another challenge is the potential for overfitting, where the model learns the noise and random fluctuations in the training data, rather than the underlying patterns and relationships. Poor performance on new, unseen data can result from this. Despite these challenges, great promise has been shown by deep learning, and it continues to be an active area of research and development. Techniques and architectures are being developed to improve the performance and efficiency of deep learning algorithms and to address the challenges of training on large and complex datasets. The potential for its use in a wide range of applications is enormous as the capabilities of deep learning continue to improve.

### 2.3.2 Neural Network.

A neural network[19] is a machine learning model inspired by the structure and function of the human brain. The network is made up of many interconnected processing nodes known as neurons that work together to tackle complicated issues such as image and speech recognition. The neurons are organised in layers, with the input layer absorbing information and the output layer providing the final result. One or more hidden layers situated between the input and output layers execute intermediate computations.

To train, neural networks first evaluate samples of known "input" and "output," building connections between those two concepts that are then saved within the network's data structure. Most neural networks are taught by computing the difference between the network's processed output and a projected output, a process known as "training." By combining this estimation error with a learning mechanism, the network then updates its weighted associations. The neural network's output grows more similar to the anticipated result with each update.

Large datasets are frequently used to train neural networks, in which the model is exposed to multiple samples of the type of data it will most likely encounter in the real world. The model dynamically modifies the strength of connections between neurons as it studies each sample to enhance its performance on the task. This training procedure enables the model to learn from the data and effectively predict outcomes on novel data. Neural networks are widely used in a wide range of applications, including picture and audio recognition, natural language processing, and even video game play. They are also an essential component in deep learning, an artificial intelligence approach that solves complicated problems by employing multiple layers of interconnected neurons.

### 2.3.3 CNN

Convolutional neural networks [19] are a form of neural network that excels at image identification and processing. They are made up of multiple linked neurons stacked in layers, with each layer performing a specialised action on the incoming data. One of the key characteristics of CNNs is the use of convolutional layers, which perform a convolution operation on the input data. This entails sliding a tiny filter or kernel over the input and computing the dot product between the filter entries and the input data at each place. This technique helps the extraction of spatial information from input data, such as edges and forms. Another significant component of CNNs is the use of pooling layers, which minimise the output of the convolutional layers.

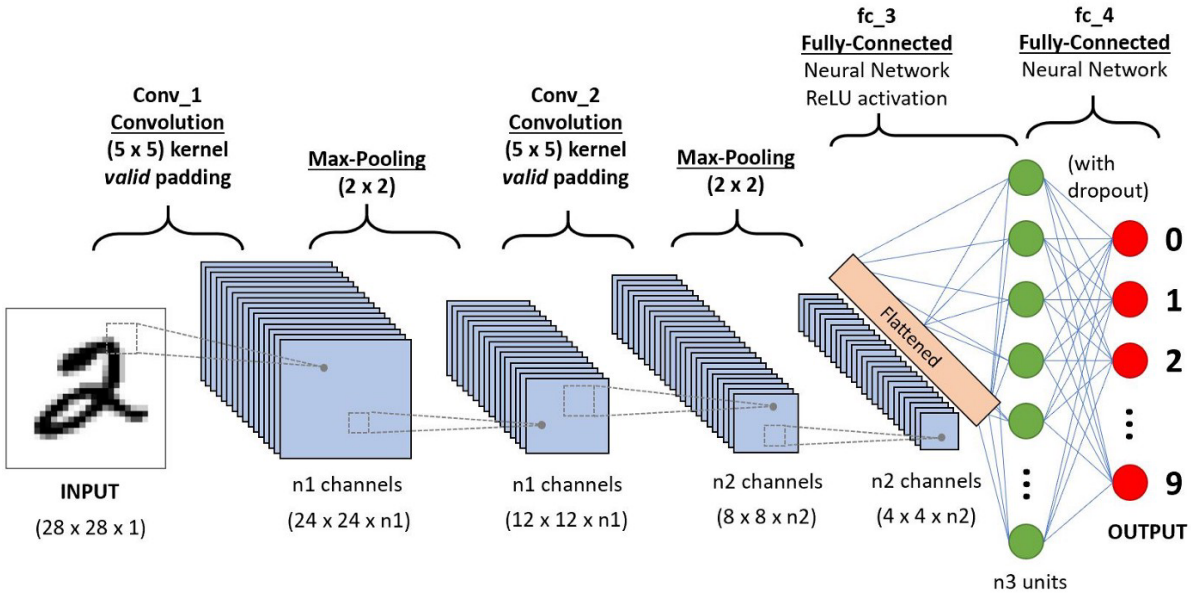


Figure 4. CNN Architecture [19]

CNNs frequently comprise additional types of layers, such as full connected layers and normalisation layers, in addition to convolutional and pooling layers. Fully connected layers, like those found in other types of neural networks, are utilised to incorporate the characteristics retrieved by convolutional layers to make a prediction or judgement. Normalization layers are used to scale and shift the input data in order to improve the stability and performance of the model.

One of the main advantages of CNNs is their ability to learn hierarchical representations of data. This mean that the model can learn to extract increasingly complex features at each layer, starting with simple features such as edges and textures in the lower layers and moving on to higher-level features such as objects and scenes in the higher layers. This hierarchical representation allows the model to make more accurate predictions and generalize better to new data.

An RNN [20] is a type of neural network that is well-suited to processing sequential data, such as natural language, time series data, and audio. Unlike a traditional FFNN, which only accepts a fixed-sized input and produces a fixed-sized output, a RNN can accept a sequence of inputs of any length and produce a corresponding sequence of outputs. The key feature of an RNN is the use of hidden states, which are like a memory that allow the model to remember and incorporate information from previous inputs in the sequence. This makes RNNs particularly effective at processing sequential data, since the hidden states can be used to capture dependencies and patterns across time. One of the main challenges with RNNs is the difficulty of training them, due to the large number of time steps and the long-term dependencies that can be modeled. This problem is often addressed using techniques such as truncated BPTT and gradient clipping. RNNs are widely used in applications such as NLP, speech recognition, and time series prediction. They are also a key component of many modern DL models, such as LSTMs[21] and GRUs.

GRU stands for Gated Recurrent Unit, which is a type of recurrent neural network (RNN) architecture. GRU was introduced in 2014 , it is a variation of the standard RNN architecture. Like other RNNs, a GRU is designed to process sequential data such as time-series data or text. It works by maintaining an internal state that captures information about the sequence seen so far, and then updates this state with each new element in the sequence. The key innovation of the GRU architecture is the use of gated mechanisms that control the flow of information

between the input, the hidden state, and the output. These gates are designed to selectively update the hidden state and selectively output information to the next step in the sequence.The use of these gated mechanisms allows the GRU to selectively store and forget information over time, making it well-suited for tasks that require modeling long-term dependencies in sequential data.

BPTT is a learning algorithm used in recurrent neural networks, which helps them learn to make predictions or generate output based on a sequence of inputs. To put it simply, BPTT helps RNNs learn by figuring out how much they got wrong and then using that information to make adjustments to the connections between neurons. This allows the network to get better and better at predicting or generating output based on a sequence of inputs. However, BPTT can be computationally expensive when dealing with long sequences, so some modifications have been proposed to make it more efficient. For example, truncated BPTT limits the number of time steps over which the error is backpropagated, while gradient clipping limits the magnitude of the gradients to prevent them from getting too large or too small.

LSTM networks are a type of neural network that can model long-term patterns in data. They work differently than traditional neural networks by using multiple memory cells controlled by gates to store and regulate information. These gates help the network to focus on the most important information and ignore the irrelevant or outdated information. This makes LSTM networks very good at tasks that require long-term memory, such as language modeling and machine translation. They can also be used for speech recognition, time series prediction, and music generation. LSTM networks are an important part of modern deep learning models.

Besides that , it can also manage sequences of variable lengths by employing a form of memory cell known as a peephole cell. This cell enables the network to access past steps and make better selections about what data to retain. LSTM networks may also learn complicated patterns over time, such as object recognition or scene recognition, by relying on lower-level characteristics such as edges and textures. This enables them to generate accurate forecasts and deal with fresh data.

### 2.3.4 Sequence to sequence

A sequence to sequence [22] network is a sort of neural network that has grown in popularity in recent years, notably for machine translation and other jobs that require creating a series of outputs from a series of inputs. Its success can be ascribed to its capacity to accommodate variable-length sequences of data, allowing it to simulate complicated input-output interactions.

The Seq2Seq network is made up of two major components: an encoder and a decoder. The encoder takes a series of inputs, such as a phrase in one language, and turns it into a context vector, which is a fixed-sized representation. This context vector summarizes the relevant information in the given input sequence and acts as a compressed representation of it.

The decoder, on the other hand, takes the context vector as input and produces a sequence of outputs, such as a translation of the sentence into another language. It does this by generating one output at a time, using the context vector and the previous outputs to inform its decisions. This iterative process continues until the entire output sequence is generated. Seq2Seq networks are often trained using a technique called teacher forcing, which involves feeding the decoder with the correct output from the previous time step during training.

This helps to ensure that the model learns to produce accurate and coherent sequences. Despite its success, Seq2Seq networks have some limitations. One of the main challenges is the difficulty of modeling long-term dependencies, as the encoder may not be able to capture all the relevant information in the input sequence. Additionally, the decoder may suffer from the problem of exposure bias, where it is trained on teacher-forced sequences during training but must generate outputs without this guidance during inference. To overcome these challenges, researchers have proposed various improvements to the Seq2Seq architecture, such

as attention mechanisms, which allow the decoder to focus on different parts of the input sequence at different time steps. These advancements have led to significant improvements in the performance of Seq2Seq networks and have made them a popular choice for a wide range of natural language processing tasks.

Text-to-speech (TTS) synthesis is one application of Seq2Seq networks, where the source sequence is a written version of the intended speech and the resultant sequence is the matching voice waveform. The encoder in TTS takes a sequence of letters or phonemes and turns them into a sequence of acoustic characteristics, like spectrograms or mel-spectrograms, which record the frequency range of the speech signal at various points in time.

These acoustic properties are subsequently sent into the decoder, which creates the matching speech waveform. Because of the unpredictability of speech signals, which can display complex fluctuations in prosody, intonation, and various other acoustic features, this approach can be difficult.

## 2.4 Advancements in TTS

Thanks to recent advancements in the area of machine learning, text to speech system has become much more advanced lately. With the help of deep learning techniques, such as neural networks, TTS models can now learn from vast amounts of speech data, making it possible to create TTS systems that require less human intervention to generate high-quality speech. One of the most prominent neural network architectures is Tacotron-2 [23].

One of the key innovations of Tacotron-2 is its use of a sequence-to-sequence model. This model takes as input a sequence of characters or phonemes, and outputs a corresponding sequence of mel spectrograms, which represent the acoustic features of speech.

In technical terms, the encoder and decoder are typically implemented using recurrent neural networks or variants such as long short-term memory. At each time step, the encoder processes the sequence of inputs one token at a time and creates a hidden state. The context vector is the ultimate hidden state. The decoder also handles the output sequence one token at a time, but it incorporates the context vector into each time step. At each time step, the decoder produces a probability distribution over all potential output tokens, and the most probable token is chosen as the output. Tacotron-2 employs a multi-stage approach comprised of two separate neural networks to produce high-quality speech synthesis. The text-to-mel network is the first network that produces mel spectrograms from text input. The waveNet vocoder, the second network, converts the mel spectrograms into a waveform that depicts the final synthesized speech.

Tacotron-2 breaks down text into individual characters before looking up the appropriate vector for each character. This generates a series of vectors representing the supplied text. These vectors are then fed into the text-to-mel network, which produces the mel spectrograms that create the synthetic speech.In the encoder input characters are represented using a learned 512-dimensional character embedding.

Character embedding[24] allows Tacotron-2 represent input text as a sequence of vectors. These vectors are then fed into the text-to-mel network to generate the mel spectrograms that ultimately produce the synthetic speech. Consider the word "salam" in Azerbaijani language to better comprehend character embedding. Each character in this word, "s","a" ,"l", "a", and "m" is given a unique vector of numbers that represents its characteristics, such as position in the word or sound. These vectors can be viewed as a "fingerprint" for each character. Character embedding is especially helpful when dealing with out-of-vocabulary (OOV) words, which are words that the machine learning model has never seen before. Tacotron-2 can better grasp the meaning of these OOV (Out of vocabulary) words and generate more natural-sounding synthetic speech by breaking them down into individual characters and assigning each character a vector.

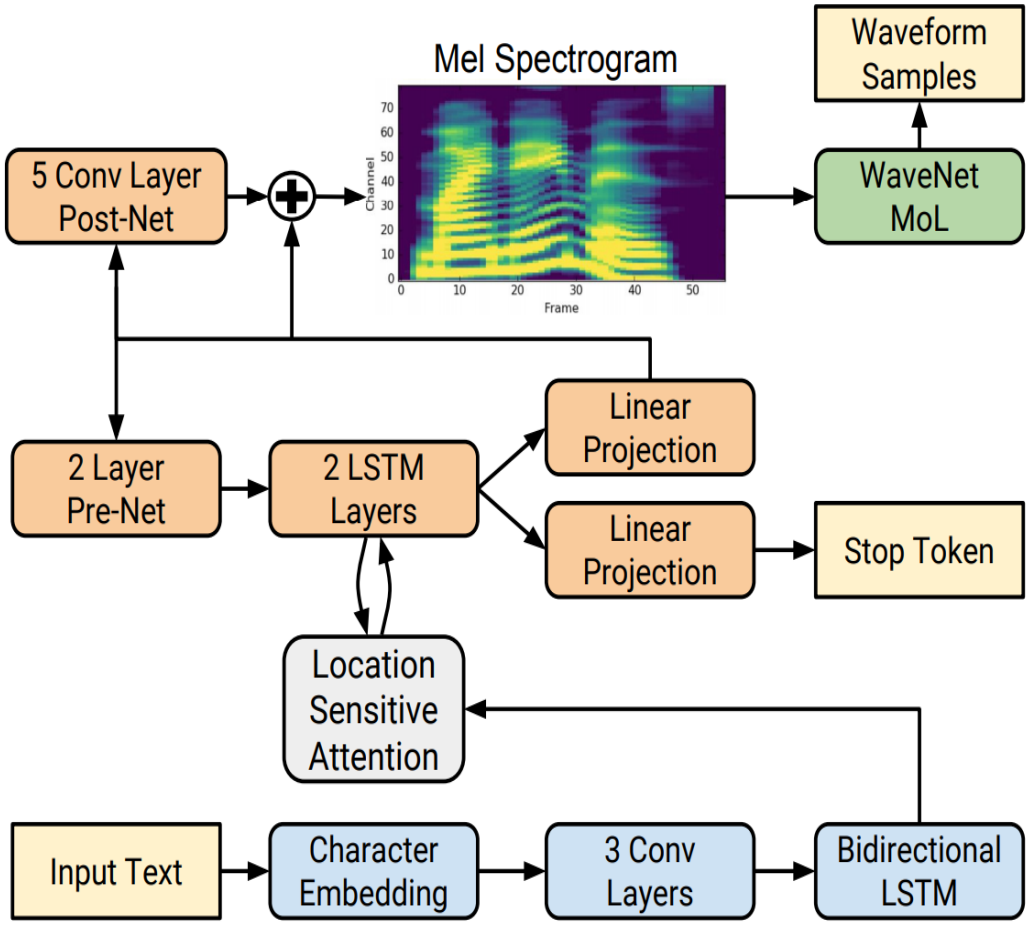


Figure 5. Tacotron-2 architecture [23]

Output of character embedding then is passed to a stack of 3 convolutional layers each with 512 filters. Then an attention network consumes the encoder output, which summarizes the entire encoded sequence as a fixed-length context vector for each decoder output phase. Normal encoder-decoder architectures have problems with decoding long sequences.

1. When the input sequence is extremely long, the fixed-length context vector will be unable to capture all of the information required to generate the output sequence and it will result in information loss.
2. Because the context vector is fixed in length, the model is forced to concentrate on the same parts of the input sequence when generating each part of the output sequence, even if those parts are irrelevant for the current output. As a result, the model may pay attention to the wrong portions of the input sequence and produce incorrect output.

The attention mechanism addresses these issues by enabling the model to focus on various parts of the sequence of inputs when producing each part of the output sequence. This means that, rather than being constrained by a fixed-length context vector, the model can listen to the most important information at each time step. By using attention, machine learning model can improve its accuracy and even handle longer inputs.

Second part of Tacotron-2 is vocoder[25]. Historically, A vocoder is referred as device that can synthesize sounds by analyzing and modifying the characteristics of a human voice or other sounds. It could be used to create a musical and vocal effects, from robotic voices to pitch correction. In this master thesis, it was decided to use Waveglow as a vocoder. Although it is possible to generate audio from only mel-spectograms just by using mathematical formulas

and algorithms such as griffin-lim, it is better to use vocoder. There are several reasons why AI-based approaches like Waveglow[26] and other vocoders systems are still needed:

- The conversion from Mel to STFT spectrogram is not entirely lossless. Firstly, the Mel spectrogram represents the frequency content of an audio signal in a non-linear scale, which is based on the human auditory system's perception of sound. In contrast, the STFT spectrogram represents the frequency content of an audio signal in a linear scale. This means that some information about the shape of the frequency content of the audio signal is lost when converting from Mel to STFT spectrogram. Secondly, the Mel spectrogram typically uses fewer frequency bins than the STFT spectrogram. This means that some of the high-frequency content of the audio signal may be lost when converting from Mel to STFT spectrogram.
- Efficiency: AI-based vocoders models like Waveglow can generate speech in real-time, which makes them useful for applications such as voice assistants and chatbots, where users expect fast and responsive interactions.

WaveGlow is a type of artificial intelligence that generates high-quality audio waveforms from spectrograms. It was created by researchers at NVIDIA in 2018 and is used for speech and music synthesis. WaveGlow works by taking a spectrogram as input then the model generates a sequence of audio samples that closely resemble the original recording. This is done through a series of mathematical operations that are designed to transform the spectrogram into an audio waveform. Generated audio can closely resemble human speech and music. This is quiet useful for a variety of applications, such as text-to-speech synthesis and music generation.For example, in text-to-speech synthesis, WaveGlow can be trained on a large dataset of recorded speech to generate new Pairs that sound like a human voice. Similarly, in music generation, the model can be trained on a large dataset of musical recordings to create new compositions that capture the style and characteristics of the original music.

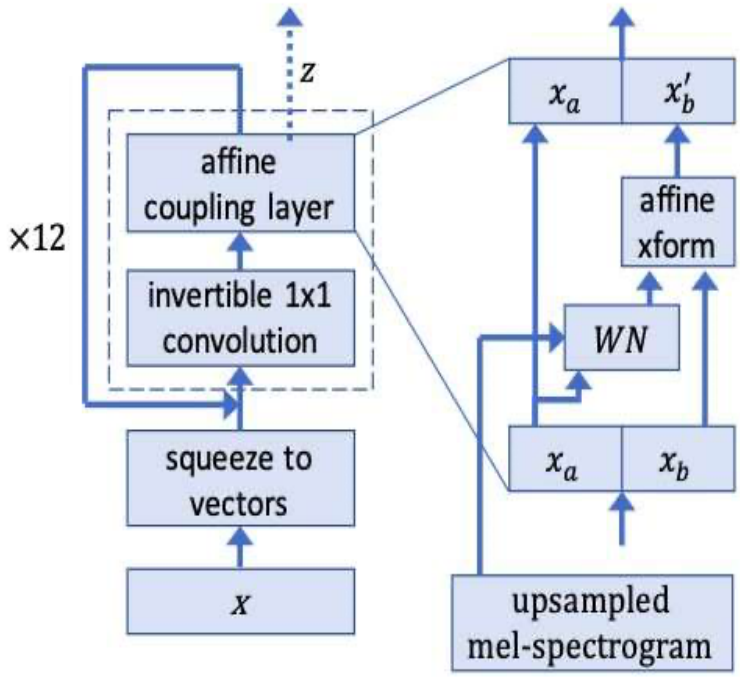


Figure 6. Wave glow architecture s[26]

The neural network used in WaveGlow has several different parts. There parts are :

1. The conditioning network, which takes in mel-spectrogram. The purpose of a conditioning network is to take in some input data, such as an image or a spectrogram, and produce a set of features or embeddings that can be used by the generative model to improve the quality of the generated output. In the case of WaveGlow, the conditioning network takes in a mel-spectrogram, which is a representation of the spectral content of the desired speech waveform. The network consists of several convolutional layers and self-attention layers that extract features from the spectrogram and generate a set of conditioning vectors.
2. Affine coupling layers. These layers take the conditioning data and use it to generate a set of transformed noise vectors, which are then combined with the conditioning data again to create the final output.
3. Invertible 1x1 convolutions. These allow the network to learn complex patterns and relationships between different pieces of data, without needing to store too much information.
4. Residual connections[27]. These are a way to help a neural network learn how to make better predictions. They work by adding the input of a layer to its output, and then the network learns to predict the difference between the input and output, instead of the output directly. This can help the network to learn more complex patterns, especially in very deep networks. In the case of speech synthesis with WaveGlow, residual connections help the network to learn how to make natural-sounding speech with a wide range of pitches and volumes. They allow the network to keep track of the small details of the speech waveform, while also learning the larger patterns of intonation and rhythm.

# 3 Research approach or methodology

This section presents an overview of the varied datasets utilized in this study, together with an in-depth analysis of the experiments performed on each dataset and the resulting outcomes.

## 3.1 Lexicon of Azerbaijani language

Azerbaijani language is a Turkic language[28] and has a rich and diverse lexicon that has been influenced by its historical and cultural background. The Azerbaijani language includes a variety of words related to different domains such as agriculture, industry, art, and science. Many words in the Azerbaijani language have their roots in Turkic languages and have been influenced by Arabic, Persian, and Russian languages. The Azerbaijani language has 9 vowel phonemes and 25 consonant phonemes, making a total of 34 phonemes. The vowel sounds in Azerbaijani are /a/, /ä/, /e/, /ə/, /i/, /ı/, /o/, /ö/, and /u/. However, if we need to differentiate between short and long vowels number of phonemes will be 40. As a comparison, Russian language has approximately 40 phonemes, Turkish language has 29 phonemes, and English language has around 44-46 phonemes, depending on the dialect. In terms of grammar, Azerbaijani is an agglutinative language, meaning that words are formed by adding suffixes to root words.

Some basic grammar rules of Azerbaijani:

1. Azerbaijani is an agglutinative language, which means that words are formed by adding suffixes to the root word. For example, the word "ev" (house) can be transformed into "evdə" (in the house), "evlər" (houses), or "evdən" (from the house) by adding different suffixes.
2. Azerbaijani has a complex system of verb conjugation, with different forms depending on tense, aspect, mood, and person. There are three tenses (present, past, and future) and two aspects (perfective and imperfective).
3. Azerbaijani has six noun cases: nominative, genitive, dative, accusative, locative, and ablative. Each case serves a different function in a sentence. For example, the nominative case is used for the subject of a sentence, while the accusative case is used for the direct object.

## 3.2 Data collection

Text-to-speech (TTS) system accuracy and robustness are crucial for enhancing the accessibility and usability of speech-based technology. In recent years, there has been a surge of interest in applying deep learning techniques to create TTS systems capable of producing high-quality synthetic speech that is indistinguishable from genuine speech. The effectiveness of these strategies, however, is strongly dependent on the availability of big and diverse datasets for training and assessment.

Obtaining high-quality and varied datasets for training TTS systems can be difficult, especially for under-resourced languages with fewer speakers and resources. As a result, there is a rising interest in creating strategies for producing synthetic speech from smaller and more constrained datasets. One possible way is to employ transfer learning, which involves fine-tuning a pre-trained model on a smaller dataset to adapt it to the target language.

Another method is to employ methods for data augmentation to broaden and expand the training dataset. Data augmentation approaches involve modifying current data to generate new instances, such as inserting noise, changing the tempo or pitch of voice, or introducing linguistic or pronunciation variances. These strategies can help to increase the TTS system's generalisation and robustness while lowering the danger of overfitting.

Despite the increased interest in building TTS systems for underserved languages, there are still several obstacles to overcome. These include the lack of standardised standards and assessment measures, the need for improved data collecting and annotation methodologies, and the need for more effective domain adaptation and speaker adaption procedures. However, with

continuing research and development, TTS systems are likely to become more accessible and effective for a broader range of languages and applications.

In the case of Azerbaijani language, despite its rich linguistic and cultural heritage, there is a lack of publicly available datasets for TTS research. The lack of data is caused by a number of factors, including the smaller number of speakers and the limited resources available for linguistic study in the language. Additionally, the process of gathering speech data for TTS systems is more expensive and complex due to the need for specialized tools, trained people, and soundproof environments. To overcome this limitation, we had to our own dataset for Azerbaijani TTS system. The dataset contains a diverse group of recordings spanning various domains, styles, and emotions. Although collecting data for TTS systems is a challenging and costly process, we believe that it is essential to develop high-quality and diverse datasets for under-resourced languages like Azerbaijani. These datasets not only contribute to the development of more accurate and natural-sounding TTS systems but also provide a valuable resource for linguistic research and language preservation.

Our data collection process involved the recording of over 6 hours of speech in total. This included the collection of approximately 2500 separate audio files, each containing a specific utterance or phrase. To ensure the quality and consistency of the data, all recordings were made by the same speaker, using the same recording equipment and soundproof studio environment. Duration of audio recorded were between 5-15 seconds. This was done to ensure that training process was faster and less error prone.

The data was collected with a sample rate of 22050 Hz, but in stereo format. Therefore, there was a need to convert the data to mono programmatically. The conversion was achieved using a Python script, which utilized the FFmpeg program. The conversion to mono helped to simplify the data and made it more suitable for use with the Tacotron-2 architecture. However, during the first attempt at training the model, it became apparent that the quality of the data collected was poor, leading to unsuccessful training. One of the main issues identified was a noticeable click sound before the speaker started to speak, and at the end of the speech. Despite efforts to programmatically remove these “click”s using the "ffmpeg" program via a Python script and converting all sounds to mono from stereo, the overall quality of the data was still insufficient. Additionally, there was a significant amount of noise in the data, and noticeable differences in volume levels between different audio files, resulting in robotic and unclear sound quality.

Given these challenges, it was determined that a different data set was necessary for successful training. After conducting extensive research, an audio book in Azerbaijani language was discovered.

This audio book was processed via speech alignment [29] tool. The practise of matching a transcription of spoken words to an audio recording of the speech is known as speech alignment. This is usually accomplished through the use of an algorithm that recognizes and matches the time of speech segments in the audio recording with corresponding words or phrases in the transcript. The purpose of speech alignment is to provide a time-stamped transcript that properly depicts the audio recording's spoken content. Forced alignment is a speech processing technique that is used to match speech recordings with their associated transcriptions or text. The process works by analyzing the audio waveform and breaking it down into smaller units of sound, such as phonemes or syllables. Then, the text transcription is split into individual words, and the algorithm matches the sound units to the corresponding words in the text. By matching the audio and text at this level, the algorithm can accurately align the audio with the transcription.

Forced alignment is a valuable tool for speech recognition and language processing tasks because it enables the automatic creation of high-quality transcriptions of spoken language. It can save significant time and effort compared to manual transcription and can be used to

process large amounts of audio data quickly and accurately. İt is important to note that forced alignment is not always 100% accurate. There are several factors that can affect the accuracy of forced alignment, including the quality of the audio recording, the quality of the text transcription, and the language and dialect being spoken. As such, it is important to carefully evaluate the output of forced alignment and make any necessary corrections or adjustments to ensure the accuracy of the transcription.

A speech recognition[30] engine is used to automatically transcribe the speech and then align the transcription with the associated audio signal in the forced alignment procedure. A language model is often used by the speech recognition engine to build a collection of candidate transcriptions for the speech signal, and each candidate transcription is then scored based on its acoustic match with the speech signal. The transcription with the highest score is then chosen and matched with the voice signal. The tool that was used is Aeneas.

Aeneas is a command-line tool and a Python library that automates the process of aligning audio and text files. It uses an algorithm to automatically align the words in a transcript with their corresponding points in the audio, creating a time-stamped file that can be used to synchronize the two. This tool is particularly useful for media professionals, as it simplifies the process of creating captions or subtitles for video content. It supports a variety of audio and text formats, and it can output the alignment results in various formats such as SRT, WebVTT, and TTML. It is designed to be highly configurable and can handle different languages and speech styles. Additionally, it has a number of advanced features such as detecting and handling overlapping speech, dealing with non-speech audio, and handling disfluencies and hesitations. Despite Aeneas lacking support for Azerbaijani language, Turkish language was employed as a substitute. Remarkably, the outcomes yielded promising results despite the occurrence of errors throughout the course of the alignment process.

After speech alignment process was done. Data set was 19 hours long and consisted of 18,000 individual audio files. The model was trained for 72 hours, and 33,500 epochs were executed. Epochs refer to the number of times the training data is passed through the machine learning algorithm during the training process. The more epochs, the more the model can learn and the better it can adapt to new data. The trained model was tested to evaluate its accuracy and reliability.

# 4 Research results and analysis of results

To assess the effectiveness and accuracy of text-to-speech (TTS) systems, several evaluation methods were developed. These methods are designed to assess the performance of the TTS system in terms of naturalness, intelligibility, and overall quality of the synthesized speech.

One commonly used evaluation method is subjective listening tests[31], In these tests, listeners are typically requested to rate the speech on a scale for dimensions such as naturalness, clarity, and overall quality. The listeners are usually given a set of instructions to follow, such as listening to each speech sample in its entirety and rating it based on the given dimensions. The speech samples can be presented in a randomized order to avoid any bias or order effects.

Another method is preference tests, the listeners are presented with a pair or set of speech samples and are asked to select which sample they prefer based on the given dimensions. This type of test can be useful for comparing the performance of different TTS systems or different configurations of the same TTS system.

In discrimination tests, the listeners are presented with a pair of speech samples and are asked to identify which sample is more natural or more intelligible based on the given dimensions. This type of test can be useful for evaluating the performance of a single TTS system by comparing different configurations or algorithms used for synthesizing the speech.

Subjective listening tests are not without flaws. The age, gender, and linguistic background of the listeners, as well as the sort of speech material utilised for testing, can all have an impact on the results. Furthermore, results may not always be constant across multiple groups of listeners, making it difficult to draw definitive conclusions.

Despite these drawbacks, subjective hearing tests continue to be an important technique for assessing the quality and usefulness of TTS systems. A full assessment of TTS system performance may be accomplished by combining subjective listening tests with various objective evaluation methods such as automated evaluation metrics and acoustic analysis. This data may subsequently be utilised to improve the quality and efficacy of TTS systems, as well as to improve the user experience.

To conduct a comprehensive evaluation of the text-to-speech (TTS) systems, a subjective listening test was performed using an online survey platform. The survey was distributed to a group of 20 participants, who were asked to evaluate 10 different audio files generated by the TTS systems. The audio files were designed to represent a range of speech styles and contexts, including conversational speech, narration, and technical readings. To facilitate the survey, a Google Form was utilized, which allowed the participants to easily and quickly evaluate the audio samples and provide their feedback. The Google Form was designed to capture a range of subjective listening scores, including naturalness, clarity, and overall quality. Participants were asked to rate each audio sample on a scale from 1 to 5, with 1 representing poor quality and 5 representing excellent quality.

When selecting phrases for evaluating a text-to-speech (TTS) system, numerous aspects must be considered to guarantee that the sentences are representative and appropriate for the assessment. For example:

- Sentence length: Select sentences of differing lengths to evaluate the TTS system's capacity to synthesise speech at various speeds and degrees of detail.
- Style and content: Choose sentences that reflect the sort of material and writing style that the TTS system is intended to handle. For instance, if the TTS
- Choose phrases with a wide range of phonemes (distinct units of sound in a language) to evaluate the TTS system's ability to create a wide range of sounds.
- Sentences that are typical of the language being synthesised: Select sentences that are representative of the language being synthesised. Choose sentences that reflect the

various accents and dialects of Azerbaijani, for example, if the TTS system is designed for usage in Azerbaijani.

- Pronunciation complexity: Select phrases including words with varying degrees of pronunciation complexity to assess the TTS system's ability to correctly pronounce words.
- Choose sentences with differing degrees of grammatical complexity to evaluate the TTS system's capacity to synthesise speech using various sentence forms.
- Topic matter knowledge: Select sentences that fall within the TTS system's topic matter expertise. Choose sentences relating to medical terminology, for example, if the TTS system is intended for medical applications.

While choosing sentences that do not have any meaning, known as "gibberish," can be useful for testing the clarity and naturalness of the TTS system's speech synthesis without being influenced by the semantic content of the text, it is important to note that this approach may not be representative of real-world usage. Users typically interact with TTS systems to convey meaningful information. On the other hand, choosing sentences that do not have logical meaning, also known as "nonsense sentences," can be useful for testing the TTS system's ability to correctly interpret and generate speech from input text. These sentences often contain grammatical structures that are technically correct but do not make sense semantically. It can help evaluate the TTS system's ability to correctly generate speech from grammatically correct but semantically nonsensical input text. However, as with gibberish sentences, it is important to also test with sentences that have meaning and are representative of real-world usage.
Example of illogical sentences in azerbaijani language are:

- “Kitab pəncərədən qaçaraq tullandı”
- “İşıq lampası mahnı oxuyurdu”
- “Səhər yeməyi asan dərslərini oxumamışdı.”
- “Ağac pişiyə hürdü.”
- “Ay düşündü ki, hava qaralıb.”
- “Kitabxana öz itini gizlətdi.”
- “Sərçə ağaca çıxıb, möhkəm qışqırdı.”
- “Yağış yağanda göbələk çətir tutur”
- “Yumşaq daşın üstündə yarpaq sürətlə yeridi"

To evaluate our system against Microsoft cognitive systems were chosen. Microsoft provides a comprehensive set of text-to-speech and speech-to-text APIs through its Azure Cognitive Services platform. The text-to-speech API allows developers to easily integrate speech synthesis capabilities into their applications, providing natural and expressive voices in over 60 languages and dialects. The system uses deep neural networks and advanced signal processing techniques to generate high-quality speech output with natural prosody and intonation. Microsoft TTS can be easily integrated into a variety of applications and platforms, including chatbots, virtual assistants, e-learning platforms, and more. It provides a scalable, reliable, and cost-effective solution for adding high-quality speech synthesis capabilities to any application or service.

Table 1:Transcript of speech samples

| # | Text |
|---|---|
| Speech Sample 1 | İşıq lampası mahnı oxuyurdu |
| Speech Sample 2 | Ağac pişiyə hürdü. |
| Speech Sample 3 | Sərçə ağaca çıxıb, möhkəm qışqırdı. |
| Speech Sample 4 | Ay düşündü ki, hava qaralıb. |

| # | Text |
|---|---|
| Speech Sample 5 | Yağış yağanda göbələk çətir tutur |
| Speech Sample 6 | Yumşaq daşın üstündə yarpaq sürətlə yeridi ki, bu hava ilə əlaqədar idi. |
| Speech Sample 7 | Kitabxana öz itini gizlətdi. |
| Speech Sample 8 | Səhər yeməyi asan dərslərini oxumamışdı. |
| Speech Sample 9 | Kitab pəncərədən qaçaraq tullandı |
| Speech Sample 10 | Onların ortaq cəhətlərə malik olması hamıya şübhəli gəlsədə hamı susdu |

Table 1 presents the transcript of ten speech samples used in the study to evaluate the performance of text-to-speech (TTS) systems. Each speech sample is assigned a unique identifier, labeled as "Speech Sample 1" through "Speech Sample 10," and the corresponding text is presented in the adjacent column.

Table 2:Subjective listening score

| # | Naturalness Sore | Clarity Score | Overall Quality score |
|---|---|---|---|
| Speech Sample 1 | 4 | 4 | 4 |
| Speech Sample 2 | 3 | 4 | 3.5 |
| Speech Sample 3 | 4 | 4 | 4 |
| Speech Sample 4 | 5 | 4 | 4.5 |
| Speech Sample 5 | 4 | 4 | 4 |
| Speech Sample 6 | 2 | 3 | 2.5 |
| Speech Sample 7 | 3 | 3 | 3 |
| Speech Sample 8 | 2 | 2 | 2 |
| Speech Sample 9 | 3 | 2 | 2.5 |
| Speech Sample 10 | 3 | 3 | 3 |

Looking at the survey results, we can observe that the ratings for naturalness are not consistent across the ten sentences. For instance, the audio for the first sentence received an average rating of 4, which indicates that the majority of the participants perceived the audio output as natural. On the other hand, the audio for the sixth sentence received a low average rating of 2.5, which suggests that the audio output was perceived as less natural by most participants.

The results depicted in the table 2 indicate that the final sentences of the analyzed corpus achieved comparatively lower scores. Upon further investigation, this trend could potentially be attributed to the syntactic complexity of the sentences in question. Indeed, studies have shown that sentence complexity can have a significant impact on the comprehensibility and readability of text. As such, it is plausible to posit that the lower scores achieved by the final sentences in the table may be a result of the increased syntactic complexity exhibited by these sentences.

Furthermore mean opinion score was calculated. MOS is a useful metric for evaluating the perceived quality of content because it takes into account the subjective opinions of real users. MOS can be used to evaluate a wide range of multimedia content, including speech, music, video, and other types of audiovisual content. By using MOS to evaluate the quality of these systems, developers can identify areas for improvement and optimize their algorithms to provide better user experiences. In our experiment mean opinion score was 3.3. This represents a fair to good quality of audio or speech. A MOS score of 3.3 falls somewhere in the middle of this scale, indicating that the audio or speech is not poor or unacceptable, but also not excellent or outstanding.

Table 3:Preference test

| Pair | Number of Listeners Who preferred Sample 1 (Tacotron) | Number of Listeners Who preferred Sample 2 (Microsoft) |
|---|---|---|
| Pair 1 | 8 | 12 |

| Pair | Number of Listeners Who preferred Sample 1 (Tacotron) | Number of Listeners Who preferred Sample 2 (Microsoft) |
|---|---|---|
| Pair 2 | 10 | 10 |
| Pair 3 | 10 | 10 |
| Pair 4 | 10 | 10 |
| Pair 5 | 9 | 11 |
| Pair 6 | 8 | 12 |
| Pair 7 | 8 | 12 |
| Pair 8 | 9 | 11 |
| Pair 9 | 7 | 13 |
| Pair 10 | 6 | 14 |

The subsequent table presents a comparative analysis of our model against Microsoft's text-to-speech system. Microsoft's platform offers two distinct voices, Banu and Babak, representing the female and male genders, respectively. Given that our model was trained utilizing a female voice, Banu was the chosen voice for comparative purposes. The results indicate that while our model exhibited superior performance for simple sentences, the complexity of the sentence positively correlated with Microsoft's text-to-speech system's higher accuracy.

The presented findings from the preference test indicate that Tacotron and Microsoft TTS possess varying degrees of merits and drawbacks. Specifically, the results demonstrate that Microsoft TTS was preferred by a majority of listeners in eight out of the ten pairs, while a tie occurred in the remaining two pairs. The evidence suggests that Microsoft TTS performed better in terms of naturalness, clarity, and overall quality, as perceived by the listeners. However, it should be noted that this preference may not hold true for all listeners or in all contexts.

Table 4: Discrimination test

| Pair | Number of Listeners Who Identified Sample 1 as more likely to be produced by human (Tacotron -2) | Number of Listeners Who Identified Sample 2 as more likely to be produced by human (Microsoft TTS) |
|---|---|---|
| Pair 1 | 8 | 12 |
| Pair 2 | 9 | 11 |
| Pair 3 | 7 | 13 |
| Pair 4 | 10 | 10 |
| Pair 5 | 10 | 10 |
| Pair 6 | 9 | 11 |
| Pair 7 | 6 | 14 |
| Pair 8 | 7 | 13 |
| Pair 9 | 8 | 12 |
| Pair 10 | 7 | 13 |

Table 4 presents the results of a discrimination test designed to evaluate the ability of two text-to-speech (TTS) systems, Tacotron and Microsoft TTS, to produce speech that is indistinguishable from human speech. The test consisted of 10 pairs of speech samples, with one sample produced by Tacotron and the other by Microsoft TTS. The listeners were asked to identify which sample was more likely to have been produced by a human. The results show that in eight out of the 10 pairs, more listeners identified the sample produced by Microsoft TTS as being more likely to have been produced by a human. In the remaining two pairs, an equal number of listeners identified each sample as more likely to have been produced by a human.

The discrimination test is a commonly used evaluation method for TTS systems, as it provides a quantitative measure of the naturalness and similarity of the synthesized speech to human speech. The results of the test suggest that Microsoft TTS may have a slight advantage over Tacotron in terms of producing speech that is indistinguishable from human speech. However, it should be noted that the test only evaluated a limited set of speech samples and that the results may not generalize to other contexts or applications. Further evaluation is necessary to fully assess the performance of the two TTS systems in a range of scenarios.

# 5 Summary and Conclusion

In recent years, there has been a growing interest in applying deep learning techniques to create TTS systems that produce high-quality synthetic speech that is indistinguishable from genuine speech. However, obtaining high-quality and varied datasets for training TTS systems can be difficult, especially for under-resourced languages with fewer speakers and resources. In this Master thesis, the development of a Text to Speech (TTS) system for Azerbaijani language was pursued. Various methods and techniques were employed to construct the system, including data collection, phonetic analysis, acoustic modeling, and speech synthesis.

While there are still several obstacles to overcome, TTS systems are likely to become more accessible and effective for a broader range of languages and applications. In the case of Azerbaijani language, there is a lack of publicly available datasets for TTS research, which necessitates the development of high-quality and diverse datasets for under-resourced languages. Our study involved collecting over 6 hours of speech data, consisting of approximately 2500 separate audio files. Despite initial attempts to train the model with this dataset, the quality of the data was poor due to various factors, such as noticeable clicking sounds and differences in volume levels. As a result, we turned to an audio book in Azerbaijani language, which was processed via speech alignment tools. The resulting dataset consisted of 18,000 individual audio files, and the model was trained for 72 hours and executed 33,500 epochs using Tacotron-2 architecture. This thesis shows that the development of high-quality and diverse datasets is crucial for the accuracy and robustness of TTS systems.

Evaluation methods have been developed to assess the effectiveness and accuracy of text-to-speech (TTS) systems, which include subjective listening tests, preference tests, and discrimination tests. These methods evaluate the performance of TTS systems in terms of naturalness, intelligibility, and overall quality of the synthesized speech. However, subjective listening tests are not without flaws and can be impacted by factors such as age, gender, and linguistic background of listeners.

Combining subjective listening tests with objective evaluation methods such as automated evaluation metrics and acoustic analysis can provide a full assessment of TTS system performance and help improve the quality and efficacy of TTS systems. In a specific instance, an online survey link was sent to 20 participants to evaluate 10 different audio files. However, the evaluation also identified areas where there is still room for improvement in terms of TTS system performance. By continuing to use both subjective and objective evaluation methods, developers can identify and address these areas to further enhance the quality and effectiveness of TTS systems.

Based on the subjective listening tests conducted with a group of 20 participants evaluating 10 different audio files, the results were fairly good in terms of naturalness, intelligibility, and overall quality of the synthesized speech. However, the evaluation also identified areas where there is still room for improvement in terms of TTS system performance. By continuing to use both subjective and objective evaluation methods, developers can identify and address these areas to further enhance the quality and effectiveness of TTS systems.

The evaluation of the TTS system also revealed that the model struggled with international words and proper names, likely due to the lack of training data for these specific cases. This highlights the importance of developing more diverse and comprehensive datasets for TTS systems, especially for languages with a diverse range of loanwords and international terms. Addressing this issue could be a focus of future work in improving the performance of the Azerbaijani TTS system.

Future work in the development of TTS systems for under-resourced languages such as Azerbaijani could involve exploring alternative data collection and processing methods to overcome the challenges posed by limited data availability. Additionally, researchers could

investigate the use of transfer learning techniques to leverage pre-trained models on larger and more diverse datasets to improve the accuracy and robustness of TTS systems for under-resourced languages. Furthermore, improving the quality and efficacy of TTS systems for under-resourced languages can have significant implications for various fields such as education, accessibility, and entertainment. Thus, future research could also focus on developing TTS systems that are tailored for specific applications such as language learning or audiobooks, which may require different speech characteristics and styles.

# REFERENCES


[1] Klatt, D (1987). "Review of text-to-speech conversion for English". Journal of the Acoustical Society of America. 82 (3): 737–93. Bibcode:1987ASAJ...82..737K. doi:10.1121/1.395275. PMID 2958525.

[2] D. O'Brien and A. I. C. Monaghan, "Concatenative synthesis based on a harmonic model," in IEEE Transactions on Speech and Audio Processing, vol. 9, no. 1, pp. 11-20, Jan. 2001, doi: 10.1109/89.890067.

[3] Titze, I.R. (1994). Principles of Voice Production, Prentice Hall, ISBN 978-0-13-717893-3.

[4] Alan W. Black, Perfect synthesis for all of the people all of the time.

[5] C. Hamon, E. Mouline and F. Charpentier, "A diphone synthesis system based on time-domain prosodic modifications of speech," International Conference on Acoustics, Speech, and Signal Processing,, Glasgow, UK, 1989, pp. 238-241 vol.1, doi: 10.1109/ICASSP.1989.266409.

[6] J.L. Gauvain, B. Prouts, C. Bouhier, R. Boesch. Generation and Synthesis of Broadcast Messages, Proceedings ESCA-NATO Workshop and Applications of Speech Technology, September 1993.

[7] van den Oord, Aäron (2017-11-12). "High-fidelity speech synthesis with WaveNet". DeepMind. Retrieved 2022-06-05.

[8] Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., ... & Saurous, R. A. (2017). Tacotron: Towards end-to-end speech synthesis. arXiv preprint arXiv:1703.10135.

[9] Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." arXiv preprint arXiv:1609.03499 (2016).

[10] Valizada, A.; Jafarova, S.; Sultanov, E.; Rustamov, S. Development and Evaluation of Speech Synthesis System Based on Deep Learning Models. Symmetry 2021, 13, 819.

[11] Abdullaeva, M. I., et al. "Uzbek Speech Synthesis Using Deep Learning Algorithms." Intelligent Human Computer Interaction: 14th International Conference, IHCI 2022, Tashkent, Uzbekistan, October 20–22, 2022, Revised Selected Papers. Cham: Springer Nature Switzerland, 2023.

[12] Gopalakrishnan, T., Imam, S. A., & Aggarwal, A. (2022). Fine Tuning and Comparing Tacotron 2, Deep Voice 3, and FastSpeech 2 TTS Models in a Low Resource Environment. In IEEE International Conference on Data Science and Information System, ICDSIS 2022 (IEEE International Conference on Data Science and Information System, ICDSIS 2022). Institute of Electrical and Electronics Engineers Inc..

[13] Abdelali, A., Durrani, N., Demiroglu, C., Dalvi, F., Mubarak, H., & Darwish, K. (2022). NatiQ: An End-to-end Text-to-Speech System for Arabic. arXiv preprint arXiv:2206.07373.

[14] Pelgrom, Marcel. (2017). Analog-to-Digital Conversion. 10.1007/978-3-319-44971-5_1.

[15] Xu, Min & Duan, Ling-Yu & Cai, Jianfei & Chia, Liang-Tien & Xu, Changsheng & Tian, Qi. (2004). HMM-Based Audio Keyword Generation. 3333. 566-574. 10.1007/978-3-540-30543-9_71.

[16] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Communications of the ACM 60.6 (2017): 84-90.

[17] Medsker, Larry R., and L. C. Jain. "Recurrent neural networks." Design and Applications 5 (2001): 64-67.

[18] Bengio, Yoshua, Ian Goodfellow, and Aaron Courville. Deep learning. Vol. 1. Cambridge, MA, USA: MIT press, 2017.

[19] Li, Zewen, et al. "A survey of convolutional neural networks: analysis, applications, and prospects." IEEE transactions on neural networks and learning systems (2021).

[20] Karita, S., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., ... & Zhang, W. (2019, December). A comparative study on transformer vs rnn in speech applications. In 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) (pp. 449-456). IEEE.

[21] Yu, Yong, et al. "A review of recurrent neural networks: LSTM cells and network architectures." Neural computation 31.7 (2019): 1235-1270.

[22] Yasuda, Yusuke, Xin Wang, and Junichi Yamagishi. "Investigation of learning abilities on linguistic features in sequence-to-sequence text-to-speech synthesis." Computer Speech & Language 67 (2021): 101183.

[23] Shen, Jonathan, et al. "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions." 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2018.

[24] Chen, Xinxiong, et al. "Joint learning of character and word embeddings." Twenty-fourth international joint conference on artificial intelligence. 2015.

[25] Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." arXiv preprint arXiv:1609.03499 (2016).

[26] Prenger, Ryan, Rafael Valle, and Bryan Catanzaro. "Waveglow: A flow-based generative network for speech synthesis." ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019

[27] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[28] The Turkic Languages", Osman Fikri Sertkaya (2005) in Turks – A Journey of a Thousand Years, 600-1600, London ISBN 978-1-90397-356-1

[29] McAuliffe, Michael, et al. "Montreal forced aligner: Trainable text-speech alignment using kaldi." Interspeech. Vol. 2017. 2017.

[30] Gaikwad, Santosh K., Bharti W. Gawali, and Pravin Yannawar. "A review on speech recognition technique." International Journal of Computer Applications 10.3 (2010): 16-24.

[31] Dybkjær, Laila, Holmer Hemsen, and Wolfgang Minker, eds. Evaluation of text and speech systems. Vol. 38. Springer Science & Business Media, 2007.