MŰEGYETEM 1782

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Telecommunications and Media Informatics

Towards Reconstructing Intelligible Speech Synthesis: An Implementation for Voice Conversion and Text-to-Speech Systems

Student: Sawalha Layan
Neptun Code: K250DO
Supervisor: Dr. Mohammed Salah Al-Radhi

Table of Content

Abstract

In our daily work, communication is needed and used in every aspect of our lives, where speech is the most used and natural way for people to communicate. This thesis discusses two comprehensive speech synthesis technologies by highlighting different approaches for it to sound as human-like as possible.

This work is divided into two parts; the first revolves around the first technology, Voice Conversion (VC). The goal of a VC system is to determine a transformation that makes the source speaker's speech sound as if the target speaker uttered it. In the typical voice conversion system, the vocoder is commonly used for speech-to-features analysis and feature-to-speech synthesis, but to avoid speech quality degradation that the vocoder might cause. A vocoder-free voice conversion approach was presented, using open-source sprocket datasets with fundamental frequency (F0) transformation to reproduce converted voices without changing linguistic features by using different datasets.

In the second part, Text-to-Speech synthesis (TTS) is introduced. TTS involves generating a speech waveform given textual input. It can be used for various purposes, e.g., car navigation, railway station announcements, telecommunications response services, and e-mail reading. This current proposal's main objective is synthesizing speech from text using recent deep learning techniques. The motivation is to create a synthetic, understandable speech that is as close to human speech as feasible. Too far, several approaches to solving this challenge have been investigated and applied.

Firstly, the framework Merlin toolkit, which is a neural network speech synthesis system, was implemented, which typically is implemented with a front-end text processor and a WORLD vocoder. But in this study, we implemented it using two different vocoders, Continuous and Ahocoder vocoders, where we investigated the different approaches for each vocoder with multiple datasets while focusing on the effectiveness of each vocoder's techniques to obtain a higher quality in TTS synthesis.

Secondly, a non-autoregressive text-to-speech model, FastSpeech2 is examined. It focuses on extracting pitch, energy and duration from speech waveform and uses them in training and interference. It was implemented to overcome common TTS issues and provide high-quality speech synthesis faster while avoiding both controllability and robustness issues. As a result, it was proved that FastSpeech2 provided the best quality among all the different approaches.

Overall, we investigated different speech synthesis approaches to produce a high-quality non-robotic human-like sound for multiple datasets. In future work, we will explore the end-to-end neural techniques to develop a data-driven Arabic TTS system. It starts by extracting pitch contour from speech waveform, refining it using the wavelet transform, and directly taking it as conditional inputs in training.

Chapter 1

Voice Conversion

1.1 Introduction

In our daily lives, communication is needed and used in every aspect of our lives, each person's unique voice remains one of the main characteristics of human speech. It's an effective way of identifying a person, even though there are many alternatives for verbal communication, but we cannot deny that it can never replace it. Speech processing can be used in various applications, such as single-channel enhancement, emotional conversion, band width extensions of narrowband speech and voice conversion [1].

Voice conversion (VC) is an important field in artificial intelligence, that belongs to speech synthesis which converts text to speech with also changing its voice identity, emotions, and accents. It is also described as converting one's voice to sound like another without changing the linguistic content. Voice conversion can identify individual voice based on its speech characteristic and substitute it with another person's voice with keeping all information and not modifying the transferred message [2]. The main objective of voice conversion is to change voice of source speaker to get a transformed signal which contains the output for target speaker. Some voice conversion applications are talking devices, vocal tools to help who have voice disorder, dubbing movies and translation to different languages.

1.2 Problem Definition

There are a variety of voice characteristics, such as fundamental frequency (F0) patterns and accents characteristics which is unique from one speaker to another; that are usually restricted by their physical constraints because of the different speech production mechanisms. Those physical constraints are important to distinguish speaker individuality and emotions but can cause barriers from producing the desired voice characteristics, if the individual speaker can get rid of these barriers, they can freely produce various voice characteristics. Voice conversion (VC) is one of the potential solutions to enable speaker to produce speech sound beyond their physical constraints.

In this chapter, we focus on the conversion of the speaker individuality, to convert source speaker to target speaker, we train a set of speech, we usually assume that the set consist similar linguistic features of both source and target speakers to model the conversion functions we use parallel dataset, several techniques such as the Gaussian mixture model (GMM), Gaussian process regression, non-negative matrix factorization, and deep neural networks have been proposed [3]. Sprocket is introduced as an open-source voice conversion software to convert voice with accuracy and individuality in the target voice which different techniques were used such as GMM, vocoder-based framework, vocoder-free framework based on GMM (DIFFGMM) and F0 transformation technique have been implemented. In Sprocket, we easily reproduce converted voiced using dataset, develop voice conversion system using parallel dataset such as acoustic feature extraction, time alignment, GMM training, feature conversion and waveform generation.

The main goal of the voice conversion system that was created is to determine a transformation that makes the speech of the source speaker sounds as it were uttered by the target by implementing both a vocoder and a vocoder-free voice conversion approach using sprocket open-source datasets. Unlike typical voice conversion systems where vocoder is used, we implemented a vocoder-free system to avoid the voice degradation that can be caused from vocoder. Our focus was to reproduce converted voice without changing any of its linguistic feature with and without the F0 transformation.

1.3 Proposed Methodology

1.3.1 Use of Sprocket

Sprocket is an open-source software that converts source speaker individuality to the target speaker using the GMM based voice conversion technique with a parallel dataset. The license of sprocket is a MIT, which is permissive free software license originating at the Massachusetts Institute of Technology that be accessed by expert and non-expert users, research, or industrial purposes. The main programming language used is Python and it is used as a Unix environment [4]. As shown in Figure 1, we begin by installing dependent libraries via the pip command. To prepare a speech dataset, we need to prepare a parallel dataset consisting of the same dialogue by different speakers, where in this experiment we trained 4 different datasets, 2 males and 2 females: BDL (US male), SLT (US female), AWB (Scottish male), and CLB (US female). The supported file formal of the speech signal for sampling rate, single channel and 16-bit waveform is 16000 Hz, 22050 Hz, 44100 Hz or 48000 Hz, where each waveform is stored in a multiple waveform every 5 seconds each as data/wav. We generate list files and configure files to initialize and use them for training and conversion processes [5].

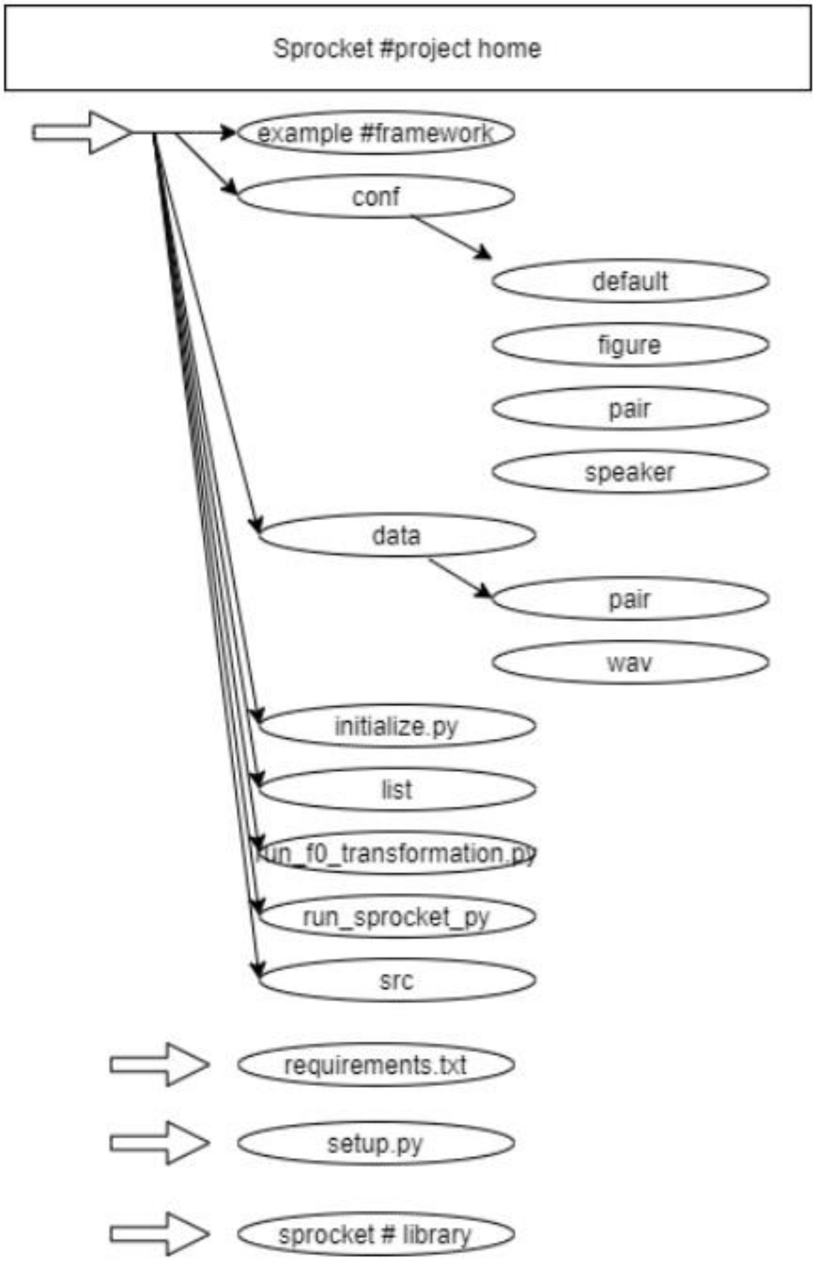


Figure 1: Structure of Sprocket.

1.3.2 GMM based Voice Conversion

In this section, voice conversion method that was used is Gaussian mixture model (GMM). GMM is a technique used to convert the voice of a source speaker into that of a target speaker by converting acoustic features such as F0 transformation. In the conversion method, there are two steps: training and conversion processes. GMM-based voice conversion uses the dataset of parallel speech of the source and target speakers, there are two types; the first one is a maximum likelihood parameter generation (MLPG) using global variance (GV) based on GMM or vocoder free voice conversion using the log-spectral differential (DIFVGC) which is used in sprocket [6].

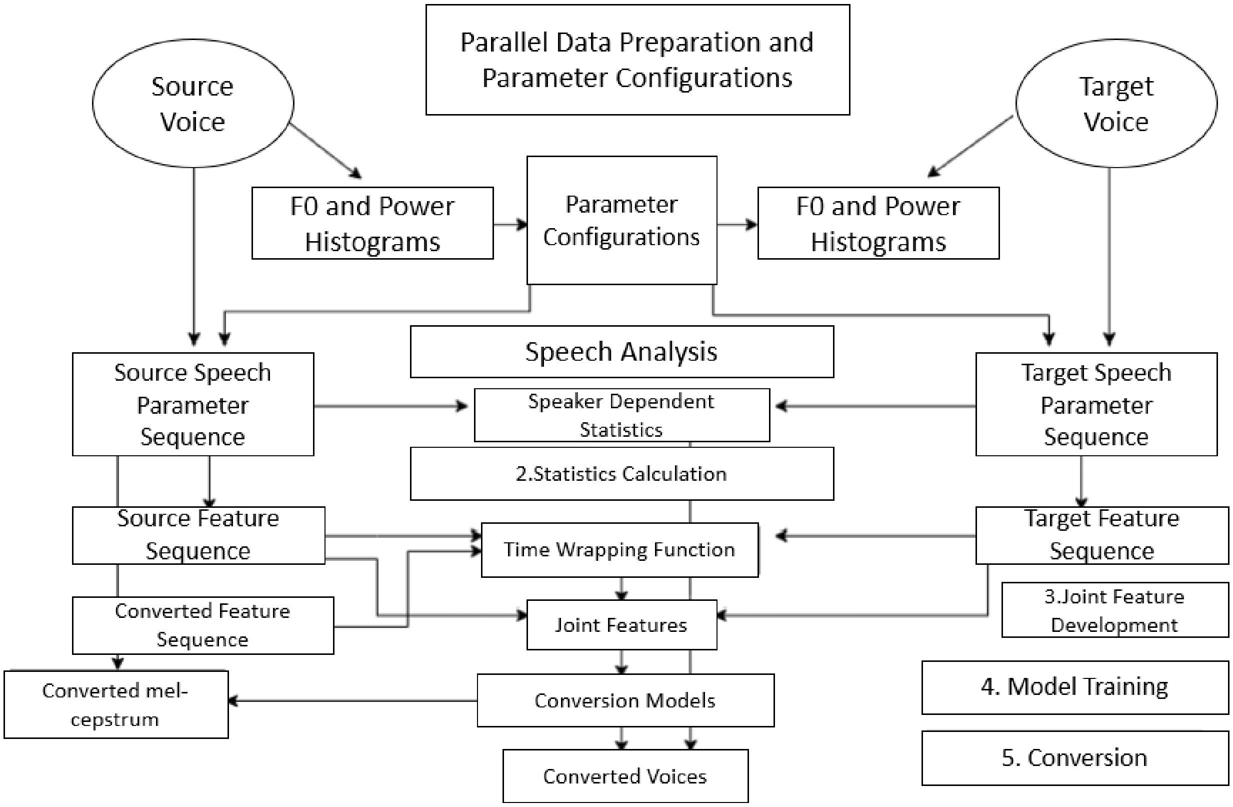


Figure 2: Training Process of the GMM-based VC method using a parallel dataset.

In Figure 2, it shows the training process of the GMM-based voice conversion method. Firstly, prepare of the parallel speech, in which we train a conversion model, which is important to prepare parallel speech with the same linguistic features and different speakers. In order to prepare a dataset, we let the source and the target speakers utter the same sentences in manuscripts, usually 50 (3-5) seconds are usually used [7]. Secondly, acoustic features, including F0 are extracted from speech signals of the source and the target speakers by sitting configuration parameters for the acoustic feature extraction process of F0 to avoid any errors in the acoustic feature extraction. In the third step, we extract acoustic features from speaker dependent statistics such as mean and standard derivation of the logarithmic F0. The next step is to model a GMM based joint probability function with frame-aligned joint feature vectors. Because of the speech signal are now always aligned due to the difference in the speaking styles between source and target Speakers [8]. To align the voices frame by frame, we prepare parallel datasets, then extract acoustic features, after that we calculate acoustic statistics features, followed by aligning the source and target features, then we do GMM trained based on the expectation-maximization algorithm using joint feature vector. By MLPG using GMM we convert the static and delta features of the source speakers into static features for the vectors of the target speakers. We then repeat the first two steps to the converted feature vectors where the time-wrapping function are refined due to the similarity in the temporal structure of both the source and target speaker individuality. After that we repeat the third step until the final iteration. The Mel-cepstrum is constructed from iterative time alignment and aperiodicity is constructed from time-wrapping function of Mel-cepstrum. Then to refine the joint feature vectors, where the joint probability density function based on the GMM is trained for the conversion process. The GV of the converted feature vectors is used to design GV post-filter. The last step is conversion process, the acoustic features of the source are converted to target speaker using the trained GMM using sprocket, F0 and the Mel-cepstrum, while the aperiodicity, speaking rate, temporal structure and source voice are retained [9]. F0, Mel-cepstrum and aperiodicity are extracted from source voice. Using the speaker dependent statistic of source and target speaker in the logarithmic space; F0 is linearly transformed frame by frame, the Mel-cepstrum is converted into target speaker after constructing the static and delta features vectors. GV post filter is applied to Mel-cepstrum due to the degraded GV of the converted Mel-cepstrum. To ensure that same waveform power for the source and target voice is produced, the zeroth order of the Mel-cepstrum is modified. Then generate the voice converted using excitation generation and the Mel log spectral approximation filter (MLSA) where F0 is transformed and Mel-cepstrum is converted [10].

1.3.3 Vocoder-free based Voice Conversion

In this project, we implemented a vocoder-free voice conversion system to avoid the degradation that is caused from typical vocoder-based voice conversion. For the DIFFVC based on differential GMM, the parameters of GMM are changed to a joint probability density of the source feature and a feature differential between the target and source features [11]. Then we calculate the converted Mel-cepstrum from the source by MLPG using DIFFGMM. In this method, the converted voice is generated from filtering the source voice using GV post filtered Mel-cepstrum differential and MLSA filter as shown in Figure 3.

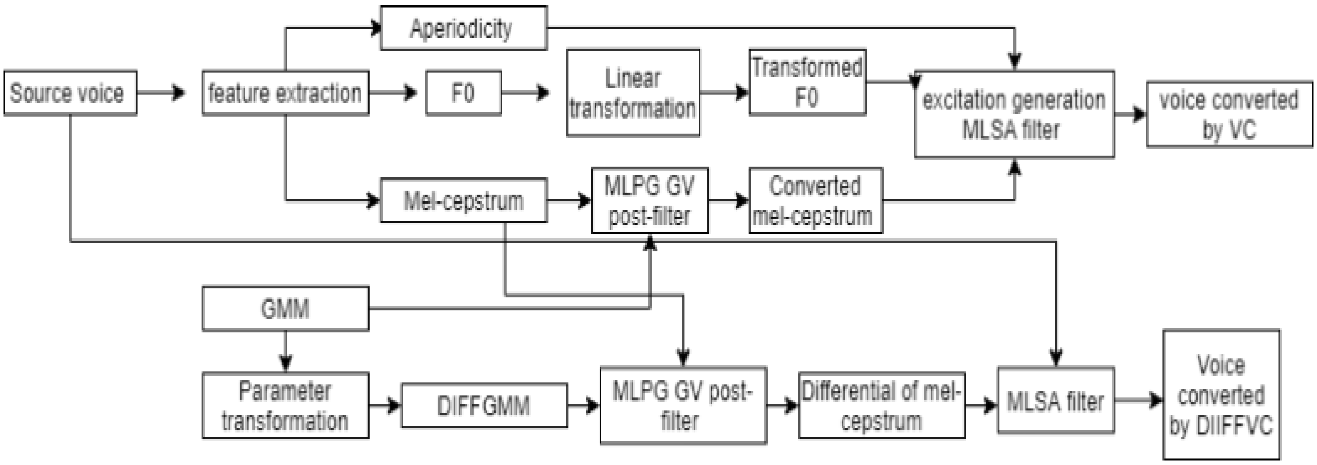


Figure 3: Conversion of the VC and DIFFVC methods.

1.3.4 F0 Transformation

Fundamental frequency (F0) is an essential acoustic feature in human speech communication and human-human interactions. F0 is a key feature for speech prosody in speech communication, it conveys linguistic information by clarifying the syntactic structure of an utterance as well as paralinguistic information such as emotions, social attitude, or speaker identity through their speaking style. This F0 transformation process changes the voice and timbre of the source voice. We apply vocoder-free F0 transformation to waveform signals to source speaker, we take advantage of the vocoder-free framework for same and cross gender conversion by the DIFFVC method. First, the F0 transformation ratio is calculated from the mean values of F0 for the source and target speakers. Then, the waveforms of the source speaker are transformed with F0 transformation ratio using duration modification techniques and resampling. For the F0 transformation technique, the waveform similarity-based overlap and add (WSOLA) method has been implemented in sprocket [12].

1.4 Results

The results of the synthesized voice conversion for BDL (US male) and SLT (US female) is divided into two parts: subjective results which are based on the preference and perceptual of the test participants; and the Objective Results are based on the actual results that was gained from the training process.

1.4.1 Subjective Results

A listening test was conducted to confirm the efficiency of our system. We compared three variants for both Female and Male speakers. In order to evaluate the converted speeches, our test participants had to listen to the source voice, vocoder-free converted speech and converted speech using F0 transformations. The test participant had to use the ACR scale. In Figure 4, we notice that the vocoder-free voice conversion was superior and incomparable to the source speaker, while the voice conversion with F0 transformation results still needs more improvement.

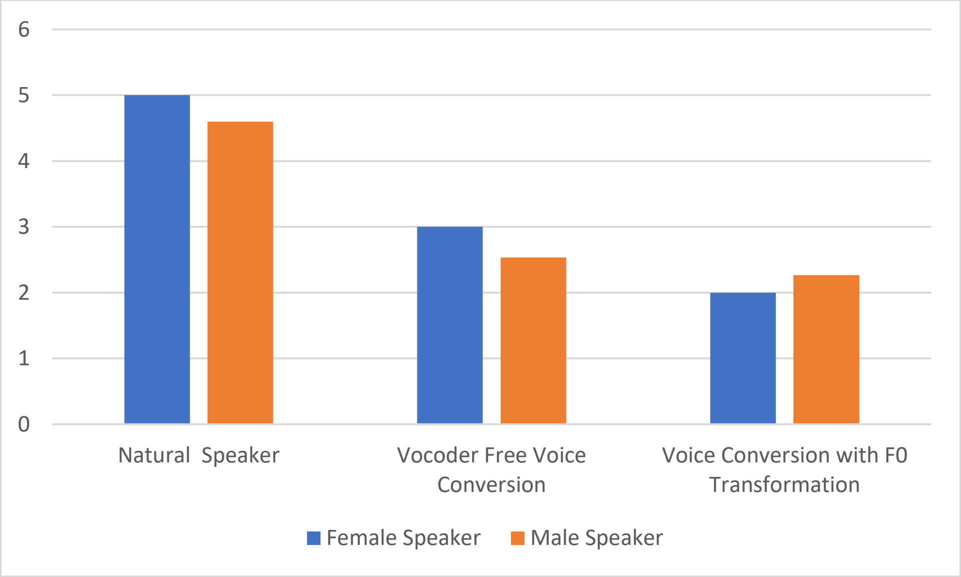
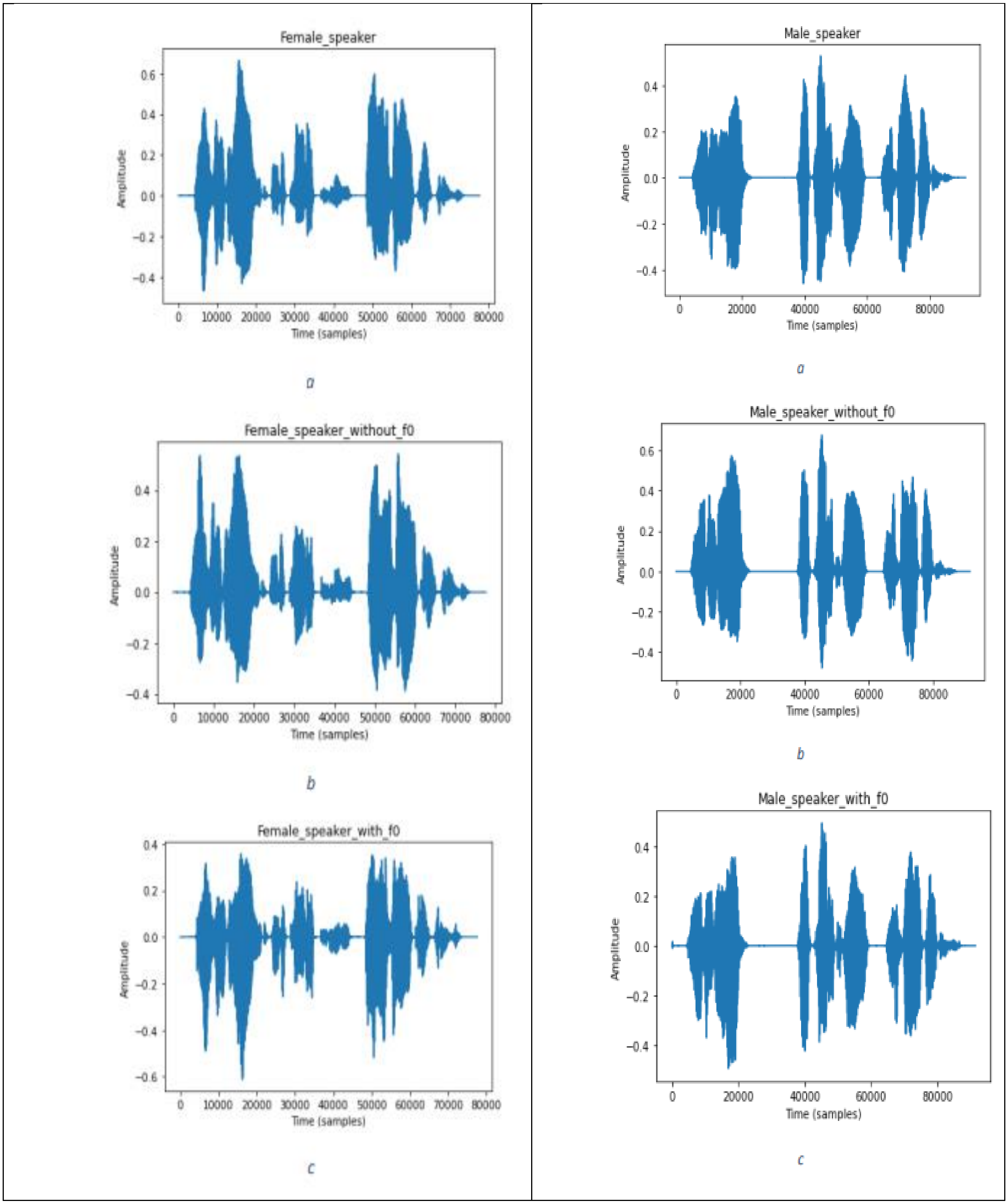


Figure 4: Voice Conversion Subjective Results

1.4.2 Objective Results

We notice that the quality for the vocoder free voice conversion system is much better than those that use a vocoder. The conversion process of the DIFFVC method with F0 transformation using waveform modification In this technique, the F0 transformation using WSOLA and resampling based on linear interpolation is directly applied to an original waveform of the source voice [13]. In Table 1, the visual representation of synthesized sounds, we describe results for female's and male's speaker original voice, after statistical voice conversion techniques based on a Gaussian mixture model (GMM) and then with a vocoder-free VC technique based on a differential GMM (DIFFVC). Since this coordinate waveform adjustment causes frequency distorting, spectral envelope too changes agreeing to the F0 transformation proportion. Hence, we ought to utilize DIFFGMM competent of changing over such a recurrence twisted source voice. We prepare the joint GMM utilizing the F0 changed source voices and the common target voices. For spectral transformation, the changed over voice is created by shifting the F0 changed source voice with changed over Mel-cepstrum differential determined with DIFFGMM inferred from the comparing joint GMM. The F0 change proportion is set to a steady value for each speaker combine. The F0 transformed source voice is generated by filtering the resulting residual signal again using the extracted Mel-cepstrum. The spectral envelope of the F0 transformed source voice is converted using the converted Mel-cepstrum differentials with DIFFGMM. We set the F0 transformation ratio to a constant value for each speaker pair. In this technique, a part of natural phase components of the source voice is well preserved because the F0 transformation is performed by directly modifying the residual signal without the vocoding process. Moreover, this technique makes it possible to freely control the F0 transformation ratio without changing DIFFGMM for the spectral differential conversion because the original spectral envelope is also preserved through the F0 transformation. We notice how results and the quality of the converted voices for those of opposite gender is much better when compared to the converted voices where the F0 transformation was not implemented. Although the DIFFVC based on the DIFFGMM makes it possible to achieve converted voice with significantly higher sound quality than that obtained by VC based on the GMM method, the conversion accuracy of the speaker similarity significantly decreases when performing the conversion for speakers with different gender (i.e., cross gender VC) because there is no F0 transformation module when using the vocoder.

Figure 5: Speech samples from Female and Male speakers using different proposed approaches.



We also extracted the Shimmer, Jitter, and spectral characteristics from the resulted voices. The different values in signal between frequency and amplitude are known as shimmer and jitter. They are the characteristics of voice signals, and they detect the roughness, hoarseness, and breathiness in the voice. Loudness of voice, consumption of alcohol, smoking or even gender affect them as well which makes it very useful to recognize familiar voices in speaker recognition systems.

To extract the Shimmer and Jitter from the produced voices in Table 1 we used the following formulas:

$$Shimmer = \frac{1}{\mathrm{N}-1}\sum_{i=1}^{N-1} |20log(A_{i+1}/Ai)|$$

$$Jitter = \frac{1}{\mathrm{N}-1}\sum_{i=1}^{N-1} |Ti - Ti + 1|$$

Jitter denotes an alternation in F0, and shimmer indicates the perturbation in the amplitude of the speech waveform.

Table 1: Objective metrics calculated on the synthesized speech from four target speakers.

Metrics	**with Vocoder Free**		**without F0 Transformation**		**with F0 Transformation**	
	Male	Female	Male	Female	Male	Female
Local Shimmer	0.082	0.119	0.099	0.165	0.087	0.082
Local Jitter	0.018	0.016	0.009	0.017	0.016	0.017
Spectral Flux Mean	0.017	0.020	0.016	0.019	0.016	0.016
Spectral Entropy Mean	4.491	4.310	4.350	4.307	3.936	3.936

Overall, the objective evaluation of the synthesized voice conversion samples showed successful outcomes by modeling the F0 transformation with a limited dataset.

Chapter 2
Text-to-Speech

2.1 Introduction

Speech Synthesis Speech synthesis is used in a wide range of applications. This technology was created to assist persons with impairments (especially the visually impaired) in their everyday lives. Because of his severe impairment, Stephen Hawking, for example, relied on a TTS to communicate with others around him. Since then, several applications have been created that are near to TTS's original value. This technology, for example, is used to produce voices to communicate messages to customers by speech, whether they are impaired, as indicated above in the context of transportation. Today, remains of TTS are quite easy to locate in our daily lives. Language translation engines are yet another example. This technique is used to advise how to pronounce the translated material to finish the textual translation [15]. In this chapter, two main speech synthesis technologies are discussed, voice conversion and text-to-speech.

Text-to-speech or TTS is a software that reads and text and converts it into speech. TTS converts any text-based message into a verbal message. TTS is an evolving field that provides faster messages with consistency, time, and money saving. You can prepare your message in text and send it as a voice, so you don’t have to record yourself, you can also make it consistent and professional by making the communications all by the same voice. TTS is beneficial to business application by assisting them in delivering a variety of notification simultaneously [16]. Here, various technologies are discussed, with highlighting its main specifications, differences, and methodologies. A neural network speech synthesis Merlin is implemented with three different vocoders to find the best voice quality.

Text-to-speech synthesis (TTS) has become a useful component in many voice applications, such as online translators and text message readers. Furthermore, TTS is nowadays available for the most widely spoken languages all over the world on the main online services. Hence, it is important to have high-quality TTS for all languages since it represents a large market with more than 300 million potential users. TTS systems have been under development for a long period and may be used for a variety of purposes. Because of the problems experienced while recording a kid and building a system to synthesize speech that sounds natural, most of the voices utilized or synthesized come from adults, or when a child's voice is synthesized, it is generally quite robotic, inexpressive, and does not sound authentic. Several approaches are currently being designed and evaluated to find at least one that meets the requirements for having a genuine human voice in a robot device [17].

2.2 Problem Definition

The artificial creation of human voices is referred to as speech synthesis (TTS). The capacity to mechanically convert a text into a spoken voice is the purpose. TTS will be built on graphemes, which are the letters and groupings of letters that transcribe a phoneme, as opposed to speech recognition systems, which employ phonemes (the smallest units of voice) to chop out sentences in the first place. This suggests that the text is the most important resource. This is normally accomplished in two stages. The first will break the text down into words and sentences and assign phonetic transcriptions to each of these groupings, as shown in Figure 5. After identifying the various text/phonetic groupings, the next step is to translate these linguistic representations into sound. To put it another way, to interpret these signals to generate a voice that will read the information. The top of the line in voice synthesis has progressed through the period, allowing four generations of Text to Speech (TTS) systems to be distinguished. From the first to the last generation, there is rule-based synthesis, concatenation synthesis, based on probabilistic speech synthesis, and machine learning. For the first three generations, the block structure is the same.

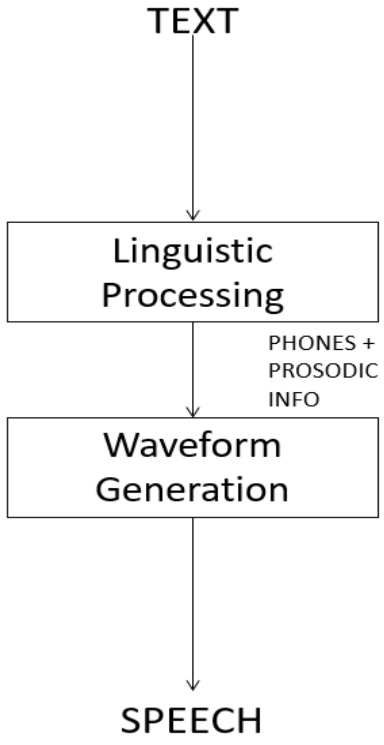


Figure 6: Block Structure for TTS systems.

To turn common language text into voice, TTS synthesis uses the creation of a speech waveform. It generally consists of two parts: a front-end and a back-end. Regularization or pre-processing, which turns abbreviations or numbers from raw user input into words, is the first of the two. The back-end is commonly a synthesizer that turns linguistic information like pitch shape and phonetic duration into voice, taking into consideration the desired prosody. The inputs for synthesis by rule are phonemes and stress marks, with a continuous waveform as the output. The approach consists of a synthesizing strategy module that includes information stored about phonemes and rules specifying the mutual effects of nearby phonemes [18].

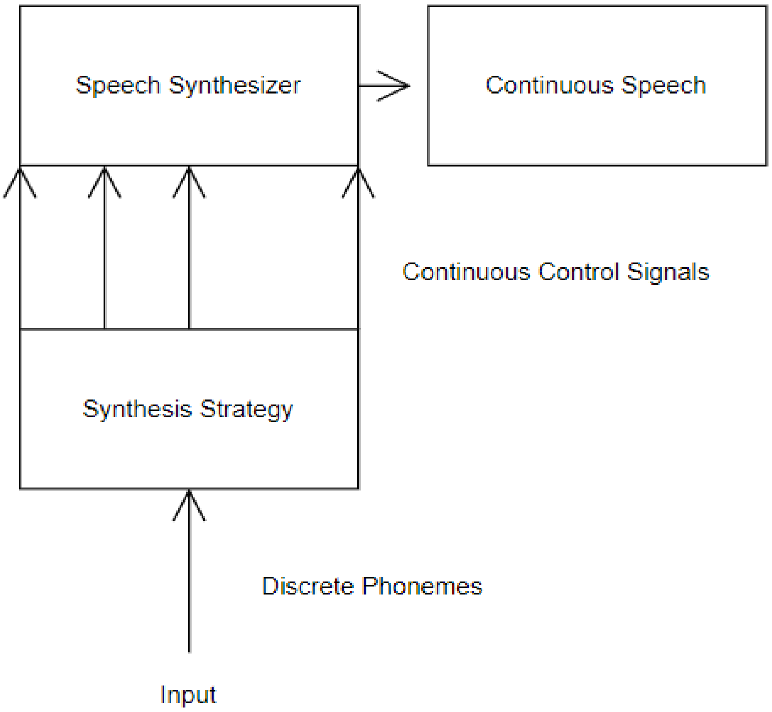


Figure 7: Technique of Speech Synthesis.

Concatenative synthesis works by capturing speech, keeping it in a database, and then concatenating the bits to get the desired result. This strategy has the potential to provide excellent outcomes. However, modeling expressiveness in speech is challenging, and strategies for autonomous waveform segmentation might result in inaccuracies in the output. A novel technique termed statistical parametric voice synthesis was created to eliminate the probable output mistakes produced by concatenation. Even though both HMMs and Deep Neural Networks are being utilized today, with the advancement of processing power and current improvements in machine learning, DNNs have begun to be employed to replace them. As deep learning progressed, writers began to incorporate neural networks into existing systems, signaling the start of the fourth generation. Later, they began to design systems entirely based on DNNs, eventually reaching end-to-end systems. A DNN is a multi-layered artificial neural network (ANN) that can model complicated non-linear interactions and build compositional models. There are several variations of this architecture. Feedforward networks including Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) are commonly used for speech synthesis applications.

When choosing the right voice synthesis, there are several factors to consider. Variables to consider include the language spoken, the type of speaker, the quality of the voice, and the provider. Now that you have this knowledge, it will be simple to pick the best solution for your goals and constraints. It's crucial to discover these partners ahead of time because not all TTS providers offer comparable product portfolios. The language and tone of voice utilized are also important aspects of the intended user experience; the voice interface and the emotions it should evoke must be in sync. Speech synthesis is based on cloud, embedded, or hybrid technology. It's worth noting that embedded has technological limitations in terms of phrase storage that a cloud does not, but the embedded voice will function regardless of what occurs when the cloud requires a connection. These characteristics should be considered based on the nature of your projects; for example, in the transportation industry, embedded is advised to assure a continuous service [19].

2.3 Methodology

In this project, Merlin was implemented. Merlin has some of the features required to build a text-to-speech system. It necessitates the use of a front-end and a vocoder, but neither is required. Data with aligned labels are also needed to train a DNN. Front-end Merlin relies on an exterior front-end, like Festival or Ossian, for DNN input. For every front-end its output must be structured as HTS-style labels, with alignment at the phone or state level. The toolbox offers routines for converting such labels into binary and continuous different feature sequences. These features are extracted from the label files using HTS-style queries, with a modest addition to allow for the extraction of continuously valued features. If the HTS-like approach isn't handy, it's also feasible to give already-vectorized input features. Vocoder STRAIGHT and WORLD are the only vocoders supported by Merlin for now. A modified version of the WORLD vocoder is included in the Merlin release, as are separate analysis and synthesizing executables. Fixed and variable frame rates (such as pitch synchronized) are supported by Merlin. Data To acquire state level aligns for the training data, HTK or HTS can be employed. Merlin may alternatively rely solely on phone level alignments, which can be determined using other tools like festvox cluster-gen. Duration modelling Merlin uses a different DNN from the acoustic model to model duration. The duration model is trained to estimate phone- and/or state-level durations on the matched data. At synthesis time, first, the duration is predicted, then the acoustic model is utilized to forecast the speech characteristics.

2.3.1 Merlin: Open-source toolkit

Merlin may be implemented in one of two ways: Limit or Full Voice. In order to download Merlin, we need the following dependencies: numpy, scipy, matplotlib, bandmat, Theano, tensorflow, sklearn, keras, and h5py. The key distinction between them is the number of utterances utilized, which is 50 in one case and 1132 in the other. Each 14 training should last 5 minutes if it is Limit and 1 to 2 hours if it is Full voice, however, this may vary depending on the system and its features. Installation Because the Merlin toolkit runs on Linux.

2.3.2 Vocoder

In recent years, the development of statistical parametric speech synthesizers and voice conversion systems has also pushed research towards vocoding techniques, in this project three different vocoders were implemented, world vocoder, continuous vocoder and Ahocoder vocoder. Vocoder is a component of various speech synthesis applications such as text to speech, synthesis, voice conversion, etc. There are different types of vocoders with similar strategies, the first stage is the analysis which is used to convert speech into parameters that presents vocal fold signal and vocal tract filter separately into the excitation signal. In the synthesis stage the parameter is used to reconstruct the original speech signal. Despite having different vocoders but the sound quality is degraded when compared to natural sound. In this project. In all types of vocoders four different types of voices (2 Females and 2 Males) which are SLT arctic, BLT arctic, AWB arctic and CLB arctic.

2.3.2.1 World Vocoder

World vocoder was used in the first part of the experience which is a free software for highquality speech analysis and synthesis. This vocoder is designed for integration systems, it estimates F0, aperiodicity and spectral envelope. WORLD is open-source speech analysis, modification, and synthesis software. It can calculate the fundamental frequency (F0) ([fundamental-frequency-estimation]), aperiodicity, and spectral envelope, as well as create speech using only estimated parameters [20]. As shown in Figure 8, it shows parameters World vocoder with Fundamental Frequency (acoustics), spectral envelope (MGC). and BAP.

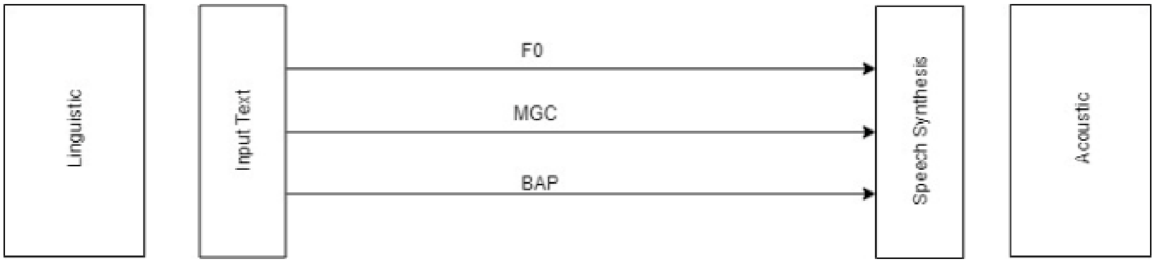


Figure 8: Parameters World Vocoder.

2.3.2.2 Continuous Vocoder

Continuous vocoder was used to overcome the shortcoming of discontinuity in the speech parameters and the computational complexity of modern vocoders. Continuous vocoder uses approach to statistical voice conversion using a feed-forward deep neural network. The approach was to integrate the continuous vocoder into the SVC framework, by converting its contF0, MVF and spectral features within a statistical conversion function [21]. The most important thing about vocoder is that it does not need to have voiced/unvoiced decision, so the alignment error is avoided between voice and unvoiced segments in SVC. As a result of its simplicity and versatility, we can build a voice converter framework with an FF-DNN. The proposed method's performance strengths and limitations for different speakers were emphasized using several metrics. In the training process, the first part is to train duration the model and the second part is train acoustic the model. As shown in the Table 2.

Table 2: Training process of continuous vocoder.

Continuous Vocoder Speech Training Results				
DNN -- MCD	MVF	RMSE	CORR	VUC
4.912	0.028	12.539	0.747	24.109%

In Figure 9, It shows the continuous parameters: Fundamental frequency (F0), maximum voiced frequency (MVF) and spectral envelope (MGC)

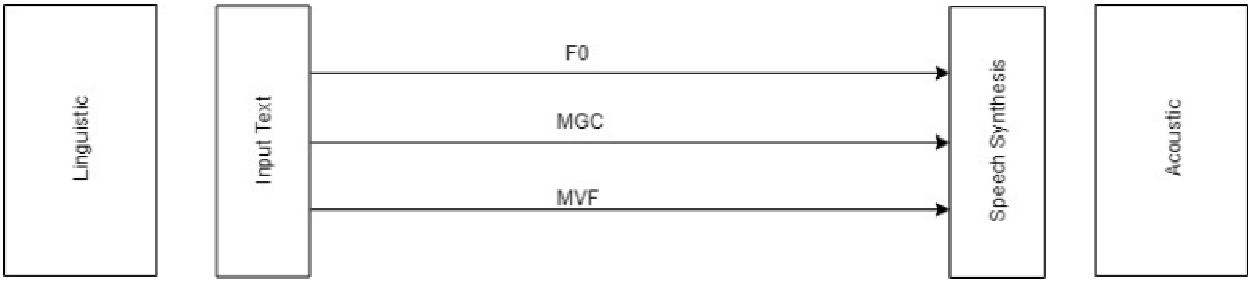


Figure 9: Parameters Continuous Vocoder

2.3.2.3 Ahocoder

The Ahocoder divides voice frames into three streams: F0, MVF, and spectrum. Both F0 and MVF are scalars: f0 can be determined by any accurate method. The method used is pitch detection algorithm returns the MVF values at the analysis frames' center. P+1 cepstral coefficients are used to represent the spectrum. [22] This distribution of Ahocoder contains two executable binary files built using gcc 4.4 under linux (64bits): ahocoder16 translates waveforms into parameters and ahodecoder16 translates parameters into synthetic waveforms. There are voiced or unspoken types, to extract their cepstral information, frames are treated differently. If the input frame was labeled as voiced, a harmonic is produced by the pitch detector. A harmonic analysis based on least squares is performed by the pitch detector. The complete analysis is subjected to squares optimization to obtain the harmonic amplitudes at various frequenciesf0 is a multiple. These amplitudes are considered distinct, even at high resolution, samples of the real spectral envelope frequencies with a low harmonics-to-noise ratio. Unvoiced frames are subjected to a quick Fourier analysis FFT, which is also known as a harmonic transform analysis with F0 equal to FFT resolution to be able to homogenize the discrete spectrum representation, the harmonic amplitudes at voiced frames provide an envelope is resampled at the FFT after being normalized in amplitude interpolation for resolution. In Figure 10, It shows the continuous parameters: Fundamental frequency (F0), maximum voiced frequency (MVF) and spectral envelope (MGC)

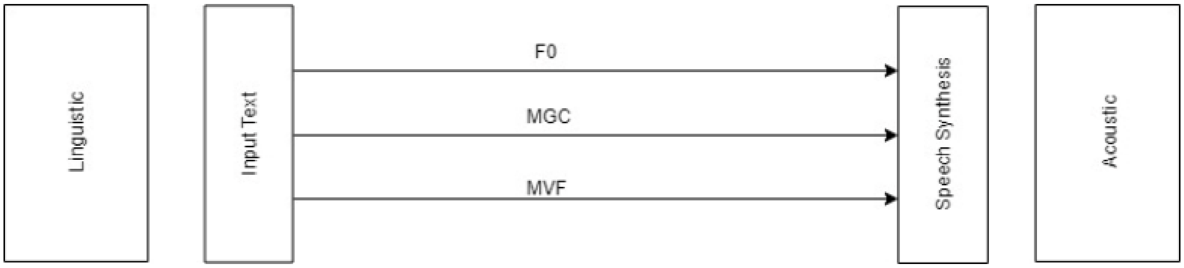


Figure 10: Parameters of Ahocoder.

During the final step of the procedure, cepstral coefficients, analysis are calculated using the amplitude the following spectral, to begin, a conventional cepstrum is obtained as follows: the log-amplitude spectrum's inverse FFT Then, the cepstrum's frequency is distorted to meet the Mel scale which describes the recursion. The Ahocoder includes linguistic processing and builds voices for some languages, such as English, Spanish, etc. The engine is acoustic, and it uses high quality vocoder.

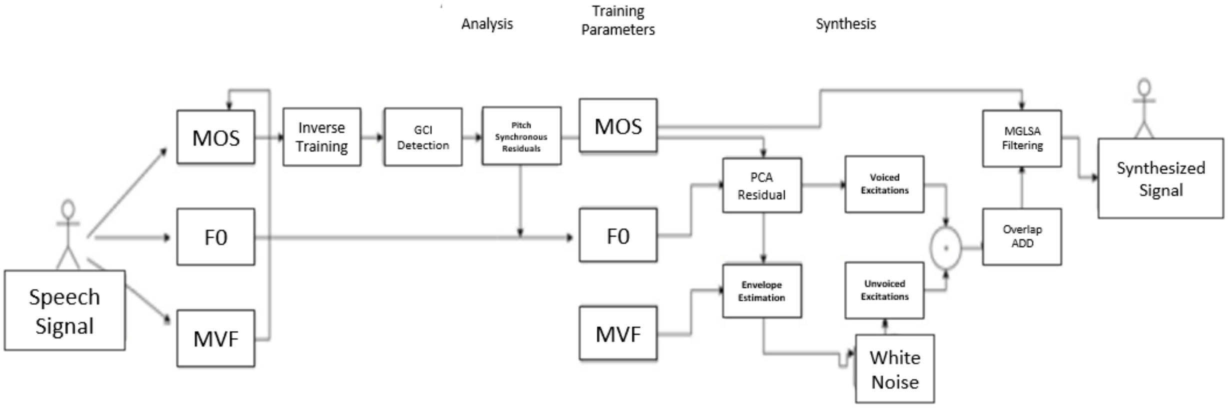


Figure 11: Workflow of the Ahocoder.

It is discovered that with HMM, a F0 used in Ahocoder produces more expressive F0 propose a new method for improving HMM-based TTS modeling piecewise F0 trajectory with voicing intensity and voiced/unvoiced decision. Proposed the F0 estimator, which is employed in this vocoder can keep up with rapid changes The technique, as shown in Figure 10, begins by separating the data [23]. Voice signal into frames that overlap each frame's windowing result is then used to the autocorrelation function should be calculated. The Kalman smoother relies on identifying a peak between two frequencies and calculating the variance to get a final sequence of values. There is no voiced/unvoiced decision in continuous pitch estimates. Furthermore, the Glottal Closure Instant (GCI) algorithm is applied throughout the analysis phase. In the vocal regions of the inversion, to find the glottal period boundaries of particular cycles residual signal filtered. A Principal Component

Analysis was performed on these pitch cycles (PCA). To produce better results, a residual is built, which will be used in the synthesis.

2.4 Results and Evaluations

2.4.1 Subjective Results

A listening test was conducted to compare the results for the different vocoder of our system. To evaluate the converted speeches, our test participants had to listen to the original voice, World vocoder voice, Continuous vocoder voice and Ahocoder vocoder voice. The test participants had to rate the quality using ACR scale. In Figure 12, we notice that the continuous vocoder was superior with a high results close to the original source, followed by Ahocoder then the world vocoder.

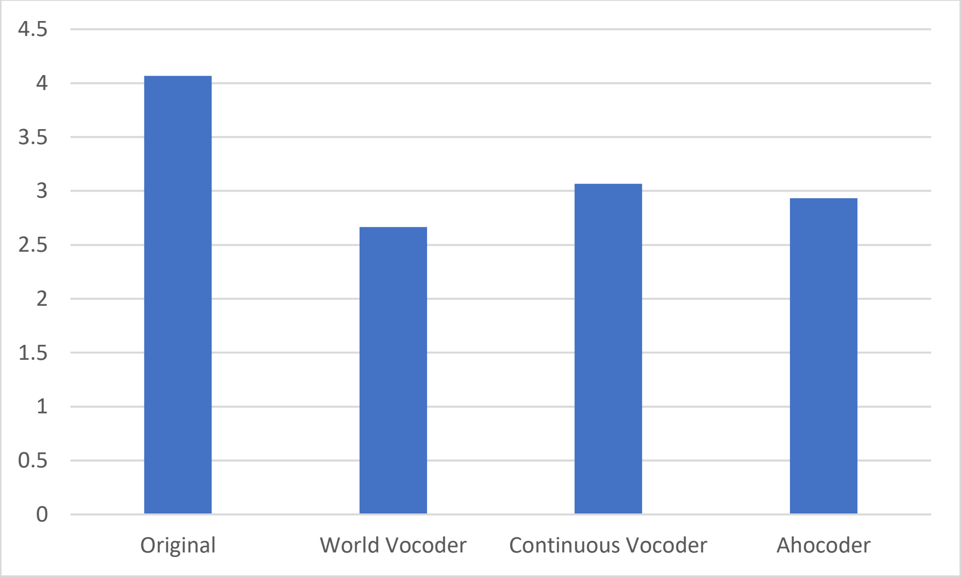


Figure 12: Sound quality of synthesized speech.

2.4.2 Objective Results

The three types of vocoders, WORLD, Continuous, and Ahocoder that were implemented are shown in Figure 13. The main goal was to integrate the Ahocoder as well as the continuous vocoder into the Merlin toolkit-based TTS by converting the contF0, MVF and spectral features in a statistical conversion function. The advantage of the continuous vocoder is that it does not need the voiced or unvoiced decision which reduced the alignment error in WORLD vocoder. While on the other hand, the Ahocoder advantage is that it provides accurate and a high quality for the speech synthesis and it is very suitable for speech manipulation and transformation. The performance of the continuous vocoder was superior in most cases to that of the WORLD vocoder and

Ahocoder. It is also proven that the impact of Ahocoder results system achieved slightly better scores than WORLD.

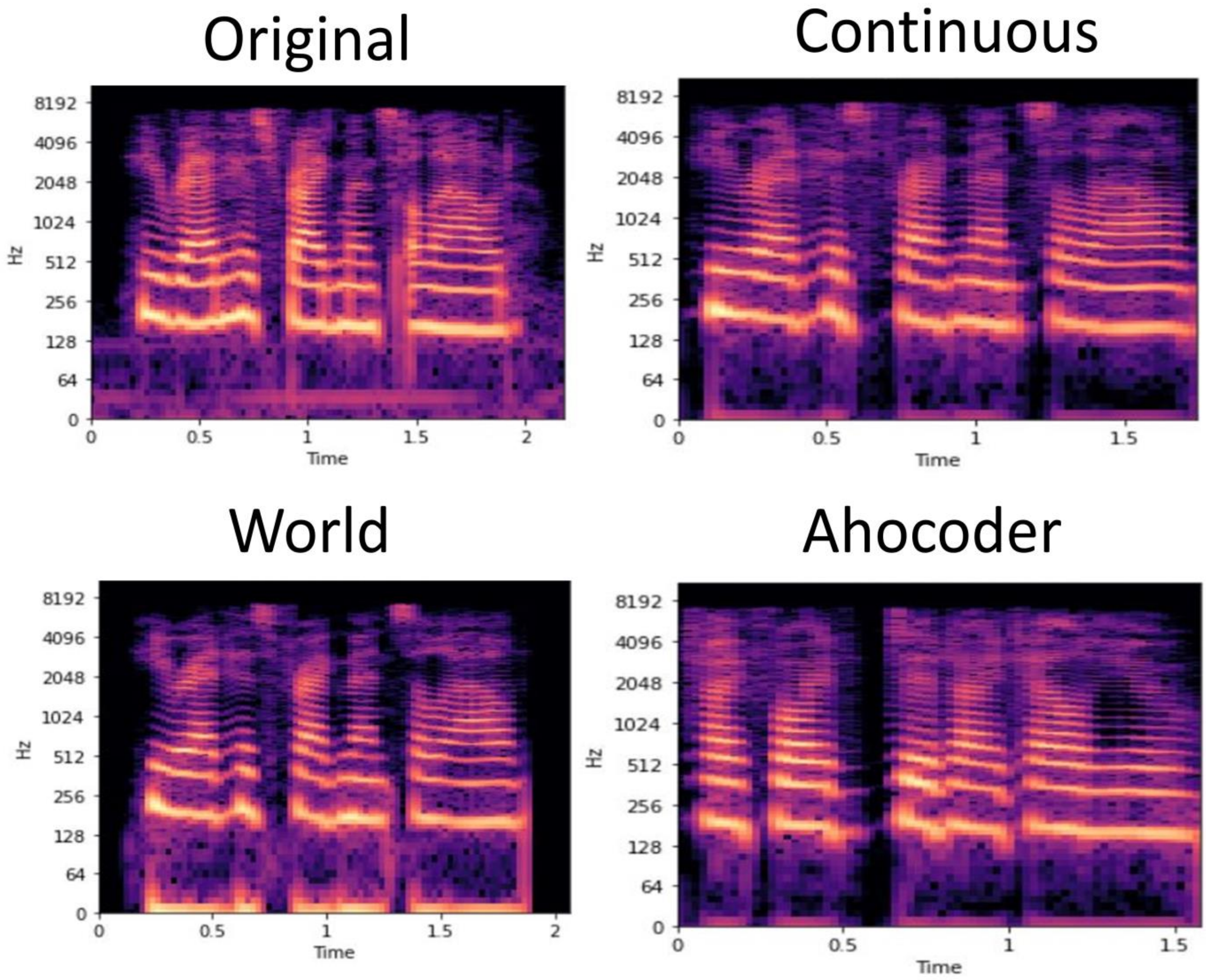


Figure 13: Results of three different vocoders.

Chapter 3

Text-to-Speech with Limited Data

3.1 Introduction

FastSpeech 2 is a neural network based on end-to-end text speech that provides high quality synthesized speech. Unlike other methods that generates Mel-spectrogram from text then synthesize speech using vocoder, to avoid slow inference speed, robustness which skips some words and lack of controllability like voice speed, fast speech is based on a transformer which generates Mel-spectrogram in parallel TTS [24].

Fastspeech 2 extracts alignment from encoder-decoder based model for phoneme duration prediction, which uses length regulator to expand the source sequence to match length of target Mel-spectrogram sequence for parallel Mel-spectrogram generation. It is named fast speech as it speeds up the Mel-spectrogram generation by 270x, end-to-end speech synthesis by 38x and it overcomes the problem of word repeating and skipping.

FastSpeech 2 also overcomes the disadvantages of FastSpeech, instead of using teacher-student distillation pipeline it directly trains the model with ground-truth target, it also introduces more variation for speech as conditional inputs. From the speech waveform, the duration, pitch, and energy are extracted and taken as conditional inputs in training where the predicted values are used in interference which results in 3x training speed over FastSpeech, it overcomes the one-to-many mapping problems in TTS and achieves better voice quality, and it can surpass autoregressive models by directly generating speech waveform from text.

3.2 Problem Definition

Text-to-Speech is often implemented or applicable in one language which most of the time is English. One of the main reasons for it is because it has a good infrastructure which in case it’s the datasets as well as its lower complexity when compared with other languages. In this project the Arabic Language is integrated, was very challenging as there is very few free speech corpora. The ultimate goal is also to be able to customize TTS to not only different language but also to different person voices with limited data as it will be so costly to collect a sufficient amount of dataset from the target voice.

In order to achieve high Text-to-Speech results, we often need a huge dataset which causes a barrier when it comes to implementing it in different languages. Nowadays there are almost seven thousand spoken languages with insufficient datasets which constraints the applicability of TTS. To get rid of these barriers we introduce Towards Reconstructing Intelligible Speech Synthesis: An Implementation for Voice Conversion and Text-to-Speech Systems.

The main goal is to implement FastSpeech2 with the English Language then to integrate another which is the Arabic Language, and then to use the same language with limited data while maintaining the fast and high-quality speech synthesis for us to make the text-to-speech synthesis applicable for more languages.

3.3 Methodology

FastSpeech2 simplifies the training pipeline and overcomes the information loss as it is trained directly by ground-truth target. Variation information of speech such as pitch, energy and accurate duration are introduced to reduce the gap between input (text sequence) and target output (Mel-spectrogram) which reduces the one-to-many mapping problem.

In the training phase the duration, pitch and energy from the target speech waveform is extracted as conditional inputs while in the inference, predicted values from predictor are jointly trained with the Fastspeech 2 model. Using continuous wavelet, pitch contour is transformed into pitch spectrogram which predicts the pitch in the frequency domain that leads to an improved accuracy of the predicted pitch [25].

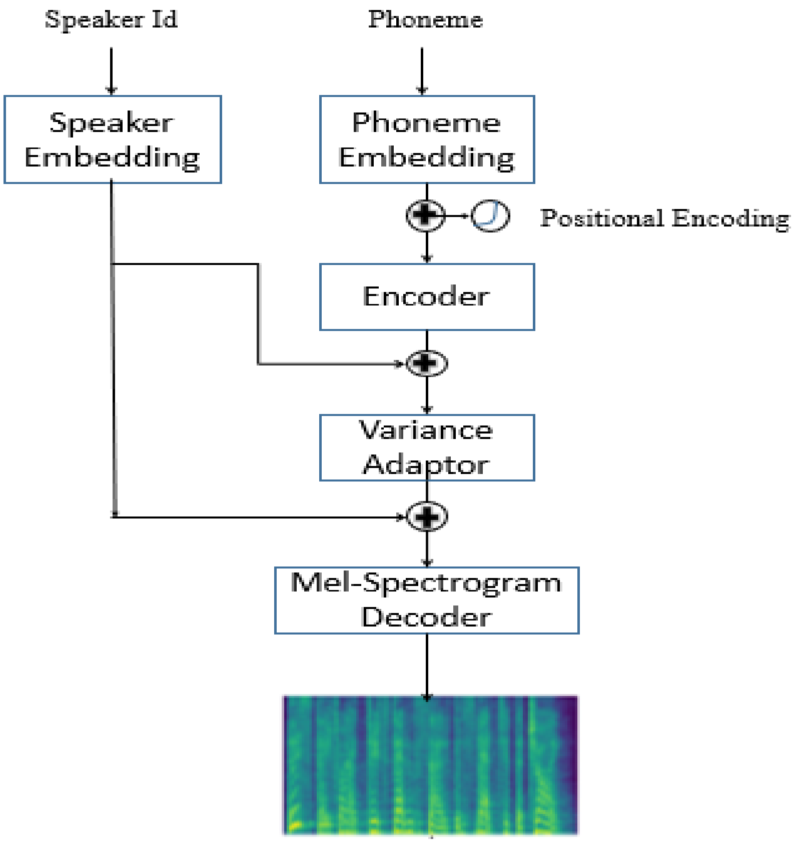


Figure 14: FastSpeech2 Architecture.

As shown in Figure 14, the representation of the architecture of FastSpeech2. Phoneme embedding is converted using the encoder to phoneme hidden sequence, where the variance adaptor adds variance information such as pitch, energy, and duration into the phoneme hidden sequence, then the Mel-spectrogram decoder converts the adapted hidden sequence into Mel-spectrogram sequence in parallel.

The ground-truth Mel-spectrograms is used for model training, which avoids the information loss and increase the upper bound of the voice quality. The variance adaptor uses the phoneme duration from the forced alignment as the training target. In the variance adaptor, variance information is added to the phoneme hidden sequence which provides information to predict variant speech. Variance information is divided into three parts, the first one phoneme duration which present the length for the speech voice sound, second one is the pitch which is a key feature that presents emotions and the last one is energy which indicates frame level magnitude of Mel-spectrograms which affects the volume and the prosody of the speech. Where the variance adaptor consists of three predictors; duration predictor, pitch predictor and energy predictor, that share similar model structure that consists of 2-layer 1D-convolutional network with ReLU activation, layer of normalization followed by dropout layer as well as an extra linear layer for the hidden states into output sequence. but different model parameters.

In training, the ground-truth value of duration, pitch and energy is extracted into hidden sequence to predict the target speech, and are also used to train the duration, pitch and energy predictors which inference to synthesized target speech [26].

The variance adaptor is divided into three parts, the first part is duration predictor which takes the phoneme hidden sequence as input and predicts the duration and represents the Mel-frame correspond to the phoneme and to ease the prediction its converted into logarithmic domain. Montreal forect alignment[27] is used to extract the phoneme duration and improve the alignment accuracy which reduces the gap for information between the input and the output. The second part is pitch predictor which predicts the variation in pitch contour where it uses the continuous wavelet transform (CWT) to decompose the continuous pitch series to pitch spectrogram and take it as training target. The last part is energy predictor, the L2-norm of the amplitude is computed using short-time Fourier transform (STFT) frame as energy, then the energy is quantized into each frame of 256 possible values. It is used to predict the original values of energy instead of quantized values.

After doing the baseline, our goal was to integrate the Arabic Language into FastSpeech2, the Arabic Speech Corpus was downloaded as a dataset. The Arabic Speech Corpus is a modern standard Arabic for speech synthesis, which contains orthographic and phonetic transcription of more than 3.7 hours of MSA speech aligned with recorder speech in the phoneme level, then the meta data was prepared and then trained the whole dataset [28]. After it was implemented, we decreased the dataset into less than a half with adjusting the FastSpeech2 parameters, encoder, variance adaptor and Mel-spectrogram decoder to match the different speaking speed, loudness, tones, and timbre, to maintain a high-quality speech synthesis.

3.4 Results

The results of the Arabic FastSpeech2, is divided into two parts, Subjective Results which are based on the preference and perceptual of the test participants and the Objective Results are based on the actual results that was gained from the training process.

3.4.1 Subjective Results

A listening test was conducted to compare the results for the Arabic Text-to-Speech with Full Data and the Arabic Text-to-Speech with Limited Data. In order to evaluate the converted speeches, our test participants had to listen to the original voice and the Arabic TTS and rate it is using ACR scale. We notice that both results are incomparable to the original sound as shown in Figure 15.

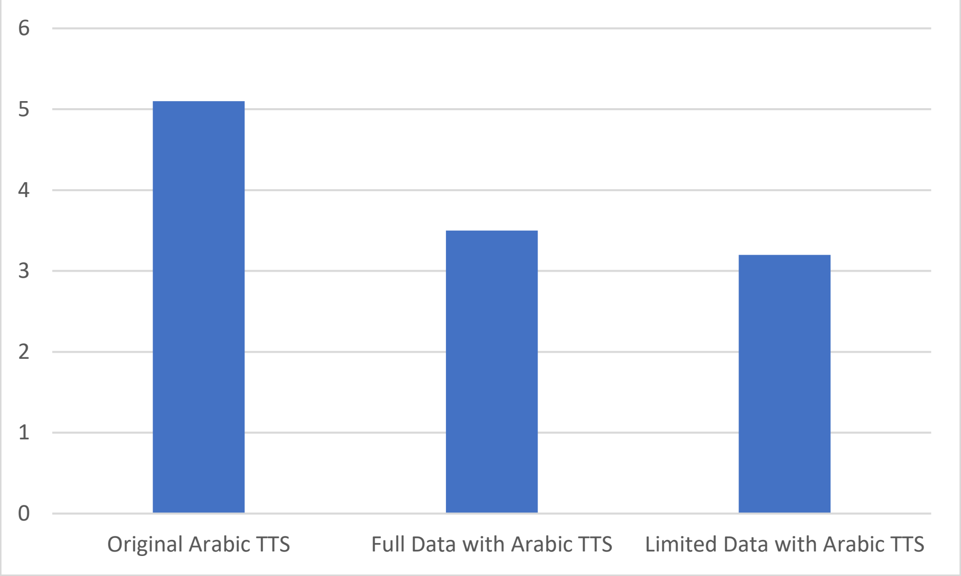


Figure 15: FastSpeech2 with Arabic Language Subjective Results.

3.4.2 Objective Results

We implemented the baseline for FastSpeech2 which only supported the English Language. We then integrated another language, Arabic Language into the system. After that we implemented it using less than half of the original dataset while maintaining high quality to be able to create a system where a user can train and generate speech using minimal dataset which will be applicable to more languages. In Figure 16, it shows the spectrogram results for the Arabic Text-to-Speech and Arabic Text-to-Speech with limited data.

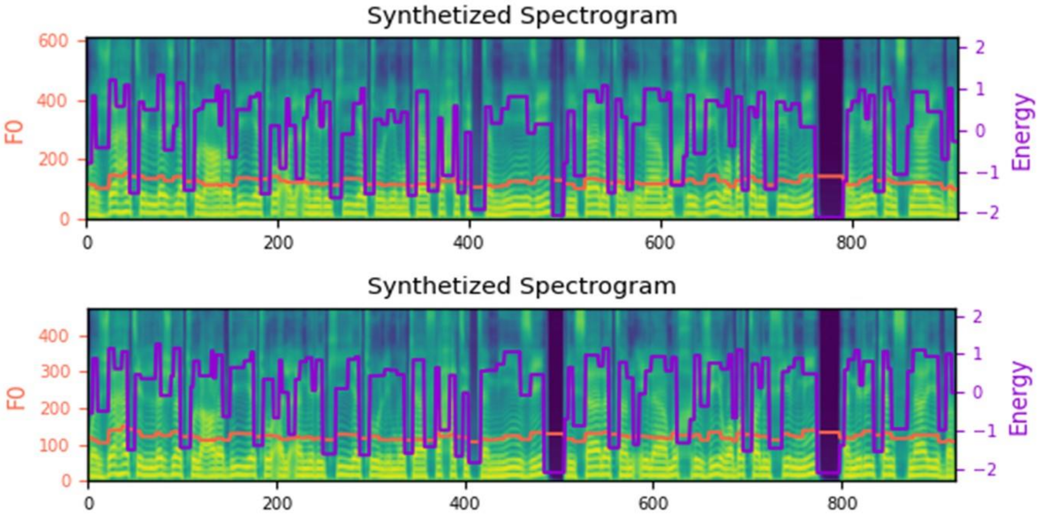


Figure 16: FastSpeech2 Objective Results with Arabic corpus: Top: Full data; Bottom: Limited data

In Table 3, we compare the results we got post training for FastSpeech2 with Arabic Language and FastSpeech2 with Limited data of Arabic Language. We notice that the result for the limited data is still incomparable to the full data synthesis.

Table 3: FastSpeech2 Full Data and Limited Data Training Results.

Metrics	Full Data	Limited Data
Mel Loss	0.473	0.549
Mel PostNet Loss	0.472	0.549
Pitch Loss	0.331	0.906
Energy Loss	0.080	0.094

Chapter 4
Conclusion and Future Work

4.1 Conclusion

In conclusion, in this research we showed two significant types of speech synthesis that focuses on testing different approaches to reach the ultimate voice quality. Both Voice Conversion and Text-to-Speech synthesis were implemented. In the voice conversion, we showed how baseline system based on “sprocket” is used for voice conversion. The baseline system consists of statistical voice conversion techniques based on a Gaussian mixture model (GMM) and a vocoder-free VC technique based on a differential GMM (DIFFVC). We investigated the effectiveness of F0 transformation techniques on 4 different datasets (2 males and 2 females), by implementing F0 transformation to obtain a higher quality voice conversion. Where the DIFFVC method using F0 transformation based on waveform modification, voice conversion performance has been significantly improved for speaker pairs whose F0 ranges are similar to each other, but the performance is still comparable to the traditional conversion method using vocoder.

While in the text-to-speech, we investigated different approaches, neural network speech synthesis system and a non-autoregressive text-to-speech model. In the neural network speech synthesis, we showed how baseline system based on Merlin which is used for Text-to-speech synthesis. We used Merlin as a tool to implement different vocoders to train different datasets (2 male and 2 female) to produce the most human-like voice. Where typically its only implemented with a front-end text processor and a world vocoder but we implemented three different vocoders to produce the highest quality for speech. We described World, Continuous and Ahocoder vocoders and how they train different datasets. We investigated the effectiveness of each vocoder’s techniques, by implementing them to obtain a higher quality in text-to-speech synthesis, which led to the conclusion that the continuous vocoder produced higher quality of speech.

In the non-autoregressive text-to-speech model we implemented Fastspeech 2 which provided not only high-quality speech synthesis but in a timely manner without controllability and robustness problems, we focused on integrating a different language into FastSpeech2 and to integrate with limited data while maintaining its high-quality produced sounds.

4.2 Future Work

In future work, we will implement voice conversion technology with another language while for the Text-to-speech we will implement it with Limited data, using person’s voice and a different language and decompose the continuous fundamental frequency (F0) with wavelet transform and take the pitch spectrogram as the training target for the pitch predictor and design an efficient neural architecture that can synthesize speech much faster in parallel.

List of Figures

List of Tables

References

[1] T. Toda, “Augmented speech production based on realtime statistical voice conversion,”Proc. GlobalSIP, pp. 755–759, Dec. 2014. [2] M. Abe, S. Nakamura, K. Shikano, and H.Kuwabara, “Voice conversion through vector quantization,” J. Acoust. Soc. Jpn (E), vol.11, no. 2, pp. 71–76, 1990.

[2] H. Doi, K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, “An evaluation ofalaryngeal speech enhancement methods based on voice conversion techniques,” Proc.ICASSP, pp. 5136–5139, May 2011.

[3] T. Toda, Y. Ohtani, and K. Shikano, “One-to-many and many-to-one voice conversion based on eigenvoices,” Proc. ICASSP, pp. 1249–1252, Apr. 2007.

[4] Toda, T., 2021. Hands on Voice Conversion. [online] Slideshare.net. Available at:<https://www.slideshare.net/NU_I_TODALAB/hands-on-voice-conversion> [Accessed 15November 2021].

[5] sprocket: Open-Source Voice Conversion Softare. 2018. [online] Available at:<https://www.isca-speech.org/archive/pdfs/odyssey_2018/kobayashi18_odyssey.pdf>[Accessed 15 November 2021].

[6] “The MIT License,” https://opensource.org/ licenses/MIT. [33] T. Toda, L. Chen, S. Daisuke, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, “The Voice Conversion Challenge 2016,” http://datashare.is.ed.ac.uk/ handle/10283/2042.

[7] Fayek, H., 2021. Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What’s In-Between. [online] Haytham Fayek. Available at: <https://haythamfayek.com/2016/04/21/speech-processing-for-machinelearning.html> [Accessed 15 November 2021].

[8] K. Kobayashi, T. Toda, H. Doi, T. Nakano, M. Goto, G. Neubig, S. Sakti, and S. Nakamura, “Voice timbre control based on perceived age in singing voice conversion,” IEICE Trans.Inf. Syst., vol. E97-D, no. 6, pp. 1419–1428, 2014

[9] S. Imai, K. Sumita, and C. Furuichi, “Mel log spectrum approximation (MLSA) filter for speech synthesis,” Electronics and Communications in Japan (Part I: Communications), vol. 66, no. 2, pp. 10–18, 1983.

[10] K. Kobayashi, T. Toda, and S. Nakamura, “Intra-gender statistical singing voice conversion with direct waveform modification using log-spectral differential,” Speech Communication, Mar. 2018 (In press). K. Kobayashi, T. Toda, and S. Nakamura, “F0 transformation techniques for statistical voice conversion with direct waveform modification with spectral differential,” Proc. IEEE SLT, pp. 693–700, Dec. 2016.

[11] K. Kobayashi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, “Statistical singing voice conversion with direct waveform modification based on the spectrum differential,”Proc. INTERSPEECH, pp. 2514–2518, Sept. 2014.

[12] Sisman, B., Yamagishi, J., King, S. and Li, H., 2021. An Overview of Voice Conversion and Its Challenges: From Statistical Modeling to Deep Learning. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, pp.132-157.

[13] W. Verhelst and M. Roelands, “An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech,” Proc. ICASSP, vol. 2, pp. 554–557, Apr. 1993. [31] k2kobayashi, “sprocket,” https://github.com/k2kobayashi/sprocket

[14] Farrús, M., Hernando, J., & Ejarque, P. (2007). Jitter and shimmer measurements for speaker recognition. Interspeech 2007. https://doi.org/10.21437/interspeech.2007-147

[15] “The MIT License,” https://opensource.org/ licenses/MIT. [33] T. Toda, L. Chen, S. Daisuke, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, “The Voice Conversion Challenge 2016,” http://datashare.is.ed.ac.uk/ handle/10283/2042.

[16] T. Toda, “Augmented speech production based on realtime statistical voice conversion,” Proc. GlobalSIP, pp. 755–759, Dec. 2014. [2] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, “Voice conversion through vector quantization,” J. Acoust. Soc. Jpn (E), vol.11, no. 2, pp. 71–76, 1990.

[17] H. Doi, K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, “An evaluation ofalaryngeal speech enhancement methods based on voice conversion techniques,” Proc.ICASSP, pp. 5136–5139, May 2011.

[18] Fayek, H., 2021. Speech Processing for Machine Learning: Filter banks, Mel-FrequencyCepstral Coefficients (MFCCs) and What’s In-Between. [online] Haytham Fayek.Available at: <https://haythamfayek.com/2016/04/21/speech-processing-for-machinelearning.html> [Accessed 15 November 2021].

[19] New Media and Mass Communication, 2022. Text To Speech Synthesis for Afaan Oromoo Language Using Deep Learning Approach.

[20] Al-Radhi, M., Csapó, T. and Németh, G., 2019. Continuous vocoder applied in deep neural network based voice conversion. Multimedia Tools and Applications, 78(23), pp.33549-33572.

[21] Erro, D., Sainz, I., Navas, E. and Hernaez, I., 2014. Harmonics Plus Noise Model Based Vocoder for Statistical Parametric Speech Synthesis. IEEE Journal of Selected Topics in Signal Processing, 8(2), pp.184-194.

[22] Erro, D., Navas, E., Sainz, I. and Hernaez, I., 2012. Efficient spectral envelope estimation from harmonic speech signals. Electronics Letters, 48(16), pp.1019-1021.

[23] Nose, T. and Kobayashi, T., 2012. Very low bit-rate F0 coding for phonetic vocoders using MSD-HMM with quantized F0 symbols. Speech Communication, 54(3),pp.384-392[1]

[24] Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z. and Liu, T.-Y. (n.d.). FASTSPEECH 2: FAST AND HIGH-QUALITY END-TO- END TEXT TO SPEECH. [online] Available at: https://arxiv.org/pdf/2006.04558.pdf.

[25] Huang, S.-F., Lin, C.-J., Liu, D.-R., Chen, Y.-C. and Lee, H. (2022). Meta-TTS: Meta-Learning for Few-Shot Speaker Adaptive Text-to-Speech. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30, pp.1558–1571. doi:10.1109/taslp.2022.3167258.

[26] Dissertation, Salah, M. and Al-Radhi, H. (n.d.). High-Quality Vocoding Design with Signal Processing for Speech Synthesis and Voice Conversion. [online] Available at: https://repozitorium.omikk.bme.hu/bitstream/handle/10890/13411/ertekezes.pdf?sequence=2&isAllowed=y

[27] McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., & Sonderegger, M. (2017). Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. Interspeech 2017. https://doi.org/10.21437/interspeech.2017-1386

[28] Halabi, N. and Wald, M. (2016). Modern Standard Arabic Phonetics for Speech Synthesis. [online] Available at: http://en.arabicspeechcorpus.com/Nawar%20Halabi%20PhD%20Thesis%20Revised.pdf