Wolwala: Deep Pashto Text-to-Speech

by

Abdul Rahman Safi

A research submitted in partial fulfillment of the requirements for the
degree of Master of Science in
Information Management

Examination Committee:	Prof. Matthew N. Dailey (Chairperson) Dr. Mongkol Ekpanyapong Prof. Phan Minh Dung
Nationality:	Afghan
Previous Degree:	Bachelor of Computer Science Kabul University, Afghanistan
Scholarship Donor:	Ministry of Higher Education (MoHE), Afghanistan

Asian Institute of Technology
School of Engineering and Technology
Thailand
July 2019

Acknowledgments

I would like to express exceptional gratitude and appreciation of mine towards my advisor, Prof. Matthew N. Dailey, for his invaluable guidance, constructive comments, and continuous encouragement throughout the period of study. His guidance has a huge impact both on my professional and personal development . Gratitude is also extended to my committee members, Prof. Phan Minh Dung and Dr. Mongkol Ekpanyapong, for providing me valuable suggestions that helped improve my research work.

I deeply thank my parents for their trust, encouragement, and patience. I am in debt of your love, care, and sacrifice you did in tough times you were going thorough due to wars and crisis but never stopped educating me. You traveled from city to city, in summers and in winters, enabling me the opportunity of acquiring knowledge. I do not remember any talk to you without this precious peace of advice: "Keep patience" which is aligned with this commandment of Allah s.w.t: "By Time, The human being is in loss Except those who believe, and do good deeds, and encourage truth, and recommend patience." I would never be able to payback a tiny portion of love and affection showered upon me.

I thank with love my wife and my baby daughter. She has helped me prepare of the dataset from the first sentence to the last one. Her companionship, support, encouragement, and help pushed me forward in the most positive way possible. I would also thank my brothers, my sister, and every member of my family for their moral support, personal attention, prayers and care.

Sincere thanks to my scholarship donor, Higher Education Development Program (HEDP) of the Ministry of Higher Education (MoHE) of Afghanistan, for providing the great opportunity to study in Asian Institute of Technology, Bangkok, Thailand. Similarly, I appreciate the efforts of every member of the department of information systems, the faculty of computer science, and the Kabul university for the coordination.

Last but not least, special credit is due to entire CSIM faculty, staff, colleagues, and friends. I thank you all for the assistance, consultation, knowledge sharing, facilitation, and support.

Abstract

Nowadays, communication plays a vital role in our highly connected world. It is practically impossible to live without information exchange. One form of information exchange is through text to speech, which benefits people in various domains and numerous applications. Text to speech combined with speech-to-text and machine translation is proven effective for cross-cultural communication, enabling an individual to communicate through his native language with speakers of other languages. Wolwala deep Pashto text-to-Speech is the text-to-speech system developed specifically for the Pashto language. Pashto is the native language of nearly 60 million people, mainly living across Afghanistan and Pakistan. The main challenges included the unavailibility of the dataset and many language-dependent components such as tokenizers, part of speech taggers, phrase breakers, and phonemic dictionary. Wolwala addressed these challenges with an unsupervised learning and some language-independent techniques. The developed text-to-speech system is based on RNN-LSTM architecture. Finaly, the system is evaluated employing both subjective and objective measures. The results were excellent. However, better results could be achieved with more data and Pashto-specific components.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Text to speech (TTS) synthesis is the artificial production of intelligible and natural-sounding speech for a given input text. Research in TTS is multidisciplinary from acoustic phonetics over morphology and syntax to speech signal processing. Figure 1.1 shows human speech production system from signal processing perspective. Despite its complexity, the research of decades has demonstrated impressive results. Recent developments has shown results that can rival human speech. This research work is the first attempt to build a TTS based on the deep learning techniques for Pastho language. Furthermore, this chapter includes the following sections: an overview of the Wolwala, problem statement, objectives, limitations and scope, and research outline

1.1 Overview

Nowadays, communication plays a vital role in our highly connected world. It is practically impossible to live without information exchange. One form of information exchange is through text to speech, which benefits different categories of people.

For example, people who access content on mobile devices are faced with small screens, making reading difficult. People who are auditory learners can use auditory learning as a complement to other types of learning styles such as visual and kinesthetic learning. This may improve the quality of learning. People with visual impairment, speech impairment, and other disabilities, people who have literacy difficulties, and people who want to multitask are a few examples. Text to speech applications are also helpful for learning new languages.

Text to speech combined with speech-to-text and machine translation is very effective for cross-cultural communication, enabling an individual to communicate through his native language with speakers of other languages. Among the languages of the Earth, English is perhaps the most the prominent language regarding text to speech, speech-to-text, and machine translation. Attempts to build such systems for other languages are in progress.

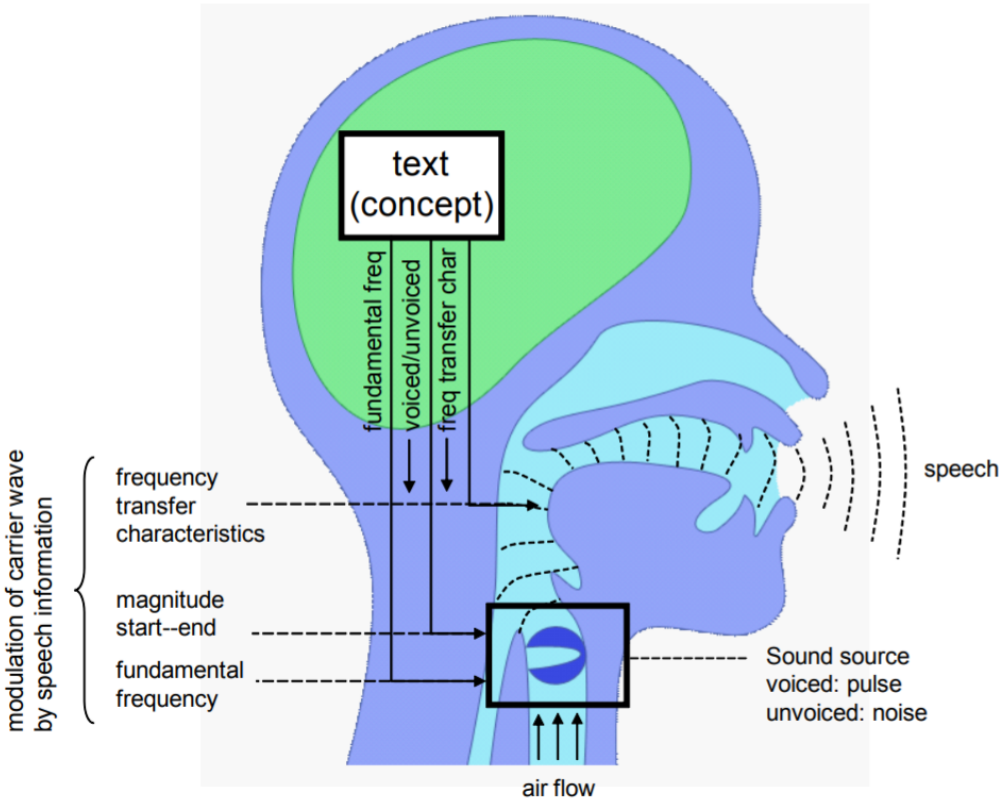


Figure 1.1: A segment of human speech production system with corresponding speech features from signal processing perspective is depicted. Each of the components plays an important role in shaping the speech signal. The main components of the human speech system are: oral or buccal cavity(mouth), nasal cavity(nose), Pharyngeal cavity(throat), larynx, tracher(windpipe), and the lungs. Normally the pharyngeal and the oral cavity are grouped into one unit called the oral tract. such as the loudness.

1.2 Progress in text-to-speech synthesis

Humans develop several capabilities throughout their lifespan. Most of these developed initially due to the need of human beings due to the needs to overcome difficult situations. Humans are intelligent social beings. Thus, communication plays vital role and they have developed various approaches of interaction among themselves. Language is one of these phenomena. Spoken form of the language is used when both sides of interaction are connected either physically or virtually through phones or other technologies. The way people communicate in spoken forms differs due to many reasons. But it can be divided broadly into two major forms: formal spoken language and informal spoken language. Formal spoken language follows the rule and regulation of that particular language. Written communication on the other hand, is the encoded version of mostly formal spoken languages. The encoding-decoding process should must be accurate. The reader must be be able to understand the

massage from the written codes.

This process is systematic and learned. It is not natural but instead developed and agreed upon. Apart form some logographic and syllabic writing systems in which signs represent words, other writing systems use graphemes to represent smaller units of sound that produces many words through their compibination.

Keeping this basic explanation in mind decoding in its basic form is a highly structured and systematic problem. However, various other aspects that are not encoded in text requires intelligence to address such as prosody.

Researchers tried many approaches. Formant and articulatory methods have been developed to address text to speech but were abandoned due to complexity of the methods and lower-quality results. Concatenatative speech synthesis in which different units of speech are stored in a database and reconstructs the utterance from the stored exemplars were developed. Concatenative speech synthesis were able to produce most natural sounds in its best cases but suffers from when the required units in similar context are missing. Furthermore, Covering all the contexts require huge huge speech databases which is almost impossible to practically to develop.

The prominent approach with many other benefits rather than storing the actual units of speech is statistical parametric speech synthesis. Statistical parametric speech synthesis learns the parameters from data. Various methods such as HMMs and different nerual network architectures such as RNNs and its variations LSTMs and BLSTMs, MDNs, and others have been applied to achieve intelligible and natural synthesized speech. Most of the text-to-speech systems are focused on intelligibility and naturalness of the synthesized speech. Prosody modeling has been the concern from the beginning of text-to-speech systems. Some transformation techniques such as voice characteristic conversion, speaker adaption, eigenvoice, and multiple regression developed has been developed. However, more research is required to address these issues.

1.3 Problem statement

Pashto (/[ˈpʌʃtoʊ]/ rarely /[ˈpæʃtoʊ]/,پښتو /[ˈpaxtō]/, /[ˈpəʂ̪ʈoː]/), sometimes spelled Pukhto, is the language of the Pashtuns.

Native Pashto language speakers are called with different variation of the same name. The variation is due to the different pronunciation of the letter "ښ" which is similar to /[ʂ]/. These variations are Pashtuns and Pakhtuns. Afghans and Pathans are two other well known variations used mostly in Indian subcontinent. Regarding the roots of the language, both Darmesteter (1888) , linguist of 19^{th} century, and Henderson (1983), modern linguist, agree that Pashto has roots in Avestan. They claim that the Rabatak inscription of Emperor Kanishka contains words from Pashto language. The inscription is written in Greek and Bactrian languages.

Another historian, Strabo (64 BC - 24 CE) affirms that the tribes living on the western Indus River lands belonged to Ariana (Strabo et al., 1917). These tribes are referred to by "Afghans or Abghan" and the language as "Afghani".

Pashto is the native language of nearly 60 million people, mainly living across Afghanistan and Pakistan. In Afghanistan it is predominantly spoken in the eastern (Nangarhar, Laghman, Kunar), central (Kabul, Logar, Wardak), southeastern (Ghazni, Khost, Paktiya, Paktika), southwestern (Kandahar, Helmand, Uruzgan, Zabul), and western (Herat, Farah) regions. In Pakistan it is primarily spoken in the Khyber Pakhtunkhwa, Tribal Areas, northeastern Balochistan, and Quetta. Apart from these native lands, native Pashto speakers have many communities spread across many countries of Asia, Europe, the Americas, and Australia due to emigration during wars and crises in Afghanistan and Pakistan; they have formed communities there.

Modern means of communication will facilitate native Pashto speakers and citizens of other nations in developing mutual understanding and collaboration in various social, economic, and educational fields. Unfortunately, not only researchers, but also giant companies such as Google, Apple, Samsung, and others have not considered the domains of Pashto text to speech, Pashto speech to text, or machine translation from Pashto to other languages yet. A translation module for Pashto exist at Google, but the quality of translation is not yet usable. It appears to be word-by-word dictionary-based translation.

Among fundamental elements (phonology, morphology, syntax, semantics, and pragmatics) across languages, text to speech applications are specifically involved with the phonological and syntactical components. Pronunciations of words in Pashto are trivial compared to English. This is mainly because the total number of Pashto letters is 45 and the total number of phonemes is 48. English, on the other hand, has 26 letters and the total number of phonemes is 44, which makes English ambiguous to pronounce.

Despite this, Pashto pronunciation is still not easy. Many factors, such as dialects, context of the word, and stress and intonation affect pronunciation. For instance, the word "مونږ, /[mu:ng]/" will be pronounced as "موژ, /[mu:ʒ]/," "مونږ, /[mu:ng]/," "موږ, /[mug]/," "مږ, /[miʒ]/" or other pronunciations but will still have the same meaning due to different dialects. The word "ښکاري, /[ʂaka:rj]/" as a noun means "hunter." But the same word as a verb means "looks like or appears." The word "غوټه" pronounced as "/[ɣu:ə]/" means "knit" and the same word pronounced as "/[ɣo:ə]/" means "diving." The word "ګټه, /[gaʈə]/" as a noun means "profit" or "stone" depending on the context. The word "سور" as a noun means "width" pronounced as "/[swr]/." But as an adjective, it means "red" pronounced as "/[su:r]/." The word "لور" has two pronunciations "/[lu:r]/" and "/[lwr]/" with three different meanings of "daughter," "sickle," and "direction."

There is a large number of Arabic words used in Pashto. Some of them are written in its original form but pronounced in a Mufaghan (the Pashto form) way such as: "ضرر, /[zarar]/," "ثواب, /[sawab]/," "صادق, /[sadəq]/." While the actual Arabic pronunciation is "/[dˤarar]/,"

"/[θawab]/," and "/[sˤadəq]/" respectively. Likewise, there is a set of Arabic letters used in Pashto that educated speakers tend to keep in their original Arabic form and pronounce them, but a common speaker may ignore them and consider the following and proceeding characters of them due to the hardship in their pronunciation for example: "عادت, /[ˈadɒt]/," and "موضوع, /[mɒːzwoː]/." In the same way, there is a set of letters which are difficult to distinguish, because they are usually pronounced in a form that is convenient to the speaker but are usually not the accurate pronunciation. Those letters are "ء، ا، ع" "ا، ح، ه," "ث، خ، س، ص" "ت، ټ، ط" "ذ، ز، ض، ظ" "پ، ف" "ک، ق" and "ګ، ږ." Technically, understanding the how and why part of these subtleties and delicacies need close collaboration between Pashto linguists and computer scientists.

Finally, I would emphasize the need to fill the existing gap in text to speech (TTS), automatic speech recognition (ASR), and machine translation domains for the Pashto language. These modules will eventually help people communicate and collaborate efficiently. My research focuses on the text-to-speech module and aims toward achieving a high-quality Pashto text-to-speech model.

1.4 Objectives

The main purpose of this research is to build a text-to-speech system for the Pashto language based on state-of-the-art techniques and recent advances. A more fine-grained set of objectives are defined as follows:

1. To prepare a reasonable-sized corpus of Pashto text and corresponding speech.
2. To develop a text-to-speech model for the Pastho language.
3. To evaluate the model based on the intelligibility and naturalness of the speech by objective and subjective evaluation measures.

1.5 Limitations and scope

The limitations and scope are as follows:

1. The scope of the model is limited to the standard dialect of Pashto languge. Other dialects will not be considered.
2. Among all possible techniques, I focus on state of the art the deep neural networks.
3. Usually, deep learning models require a large amount of data, but since this is an individual effort, and preparing an extremely large amount of data is not feasible,

I will prepare a reasonable sized corpus for research purposes. The dataset will be mainly recorded, extracted, and normalized from news broadcasts on BBC, VOA, and De Azadi Radio.

1.6 Research outline

I organize the rest of this research as follows.

In Chapter 2, I provide a a literature review of text to speech synthesis and its various techniques.

In Chapter 3, I propose the methodology that I pursue to implement text-to speech for Pashto language.

In Chapter 4, I describe the text-to-speech model development and analysis.

Finally, in Chapter 5, I provide a conclusion and recommendations for further research and improvements.

Chapter 2

Literature Review

This chapter provides a brief overview of text to speech, its history, and some of its applications. Secondly, a brief introduction to the traditional approaches used to solve text to speech problem are provided. these approaches include: articulatory speech synthesis, formant speech synthesis, concatenative speech synthesis, statistical parametric speech synthesis, and hybrid speech synthesis. Finally, state-of-the-art deep learning approaches to solve text to speech problem are discussed. The main topics in this section are: deep learning in text to speech synthesis, and end-to-end deep models for speech synthesis

2.1 Text-to-speech synthesis

Text to speech is sequence to sequence regression problem in its generic form. Given a text input, a text- to-speech system must be able to produce corresponding speech as its output. The process has many variables involved which makes the overall process complex. Despite the decades of research and investigation, generating natural speech from text is still a challenging task.

Different techniques have dominated the field over its historical timeline such as articulatory, formant, concatenative, and statistical parametric speech synthesis. Recently, Deep learning models are in hype in many domains and text to speech is not an exception. We will also look into some state-of-the-art deep models at the end of this chapter.

2.2 Text-to-speech synthesis from a historical perspective

Attempts to emulate human speech through different types of kits have a long history which dates back to thousands of years. A brief history of these attempts form mechanical to digital speech synthesis systems will give an intuition of the approaches used throughout the history to mimic the human speech.

2.2.1 Early attempts of mechanical speech synthesis

The recorded efforts of building speech synthesizers dates back to eighteenth century. As it was the era of industrial revolution, scientists tried to build machines that could generate synthesized speech.

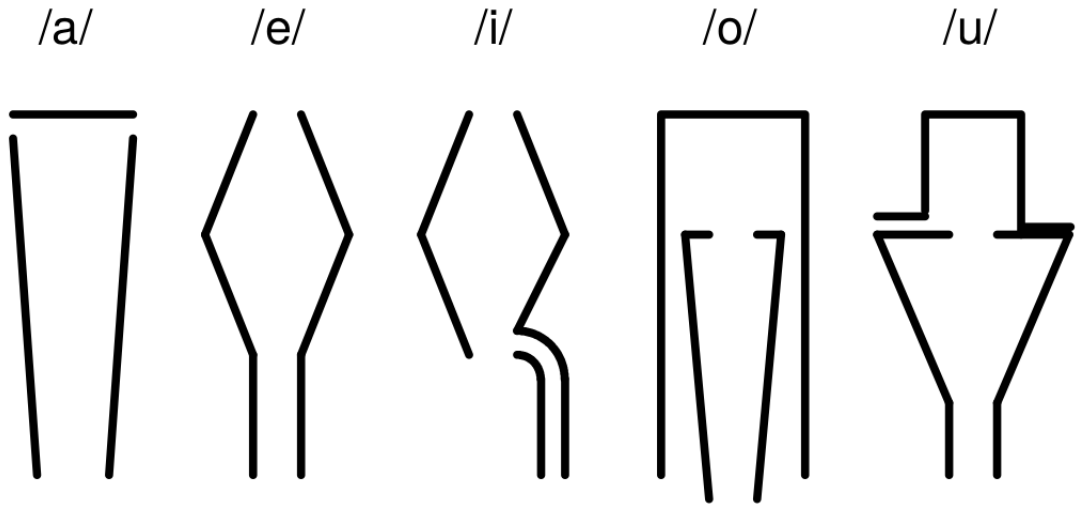


Figure 2.1: Resonators of Professor Christian Kratzenshtein. Reprinted from Schroeder (1993).

In Saint Petersburg 1779, Christian Kratzenshtein was the first person who was able to explain differences between five long vowels and made instruments to produce them artificially. The Figure 2.1 illustratess the structures made for each long vowel sound.

In Vienna 1791, a machine that could produce basic and some of the combined sounds was built by Wolfgang von Kempelen (Klatt, 1987; Schroeder, 1993).

Another machine called speaking machine, as shown in the Figure 2.2, was built by Charles Wheatstone in mid 1800's. Speaking machine could produce simple words and some combinations apart from the basic vowel sounds and most of the consonants. It was, infact, a variation or Wheatstone's version of Kempelen's machine.

Many other experiments were carried out by the scientists with the same mechanical approach to model the actual vocal tract until 1960s, but no remarkable success. Those experiments are well described in Flanagan (1973) and Schroeder (1993).

2.2.2 Era of electrical synthesizers

After the mechanical era, the next wave was of electricity. The mechanical machines had some small achievements in text speech synthesis but with electrical evolution new techniques emerged and changed the way of approaching problems.

Researcher in text-to-speech synthesis domain also tried new methods to generate synthesized speech.

In 1922, Stewart introduced a device, considered as the first electrical synthesizer, that could

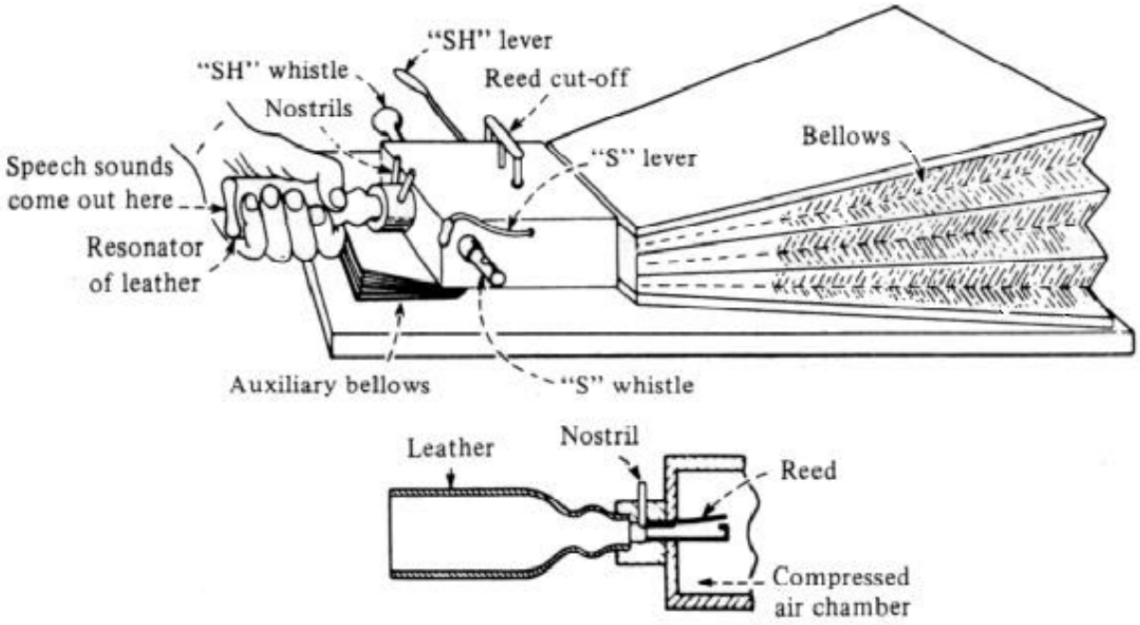


Figure 2.2: Wheatstone's speaking machine. Reprinted from Flanagan (1972).

could only generate vowel sounds. From the architectural perspective, It had two resonant circuits and a buzzer (Klatt, 1987).

In 1939, Homer Dudley, inspired by VOCODER (Voice Coder), made VODER (Voice Operating Demonstrator). Scientists produced synthesized speech that was intelligible for the first time. VODER is believed to be the first electrical speech synthesizer. If analyzed carefully, the structure of VODER is similar to source filter models (Klatt, 1987; Schroeder, 1993).

In 1951, a new device called pattern playback synthesizer was introduced by Franklin Cooper and his fellows at Haskins Labs. This device was able to convert spectrogram patterns into sounds.

In 1953, the first formant synthesizer was developed by Walter Lawrence. This development was starting point of formant synthesizers led to the emergance of many formant synthesizers.

In 1958, DAVO was created. It was the first articulatory synthesizer developed by George Rosen at MIT. Later, in mid 1960s, experiments with Linear Predcitive Coding (LPC) were made. The quality was poor. However, with some adjustments to the initial model, improvements were achievable.

In 1968, the first articulatory text-to-speech synthesizer for English language was created by Noriko Umeda and his team at Electrotehnical Lab, Japan. The synthesized speech was monotonous and low quality. However, it was intelligible.

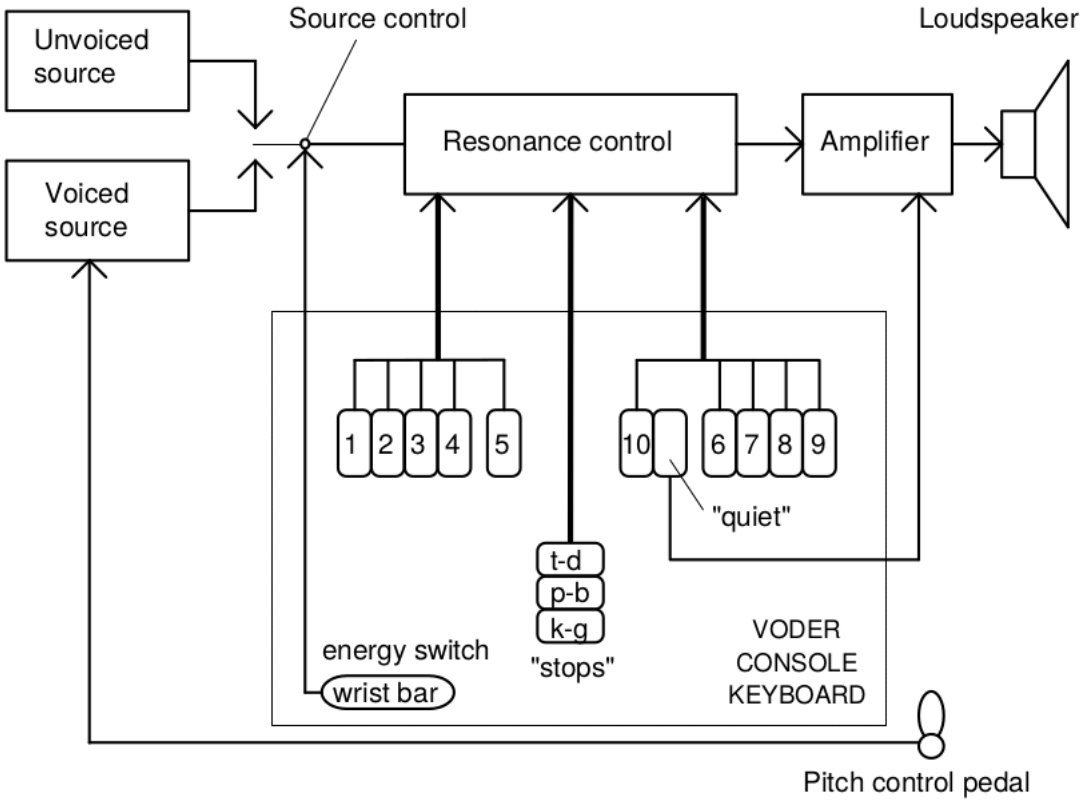


Figure 2.3: The VODER speech synthesizer. Reprinted from Klatt (1987).

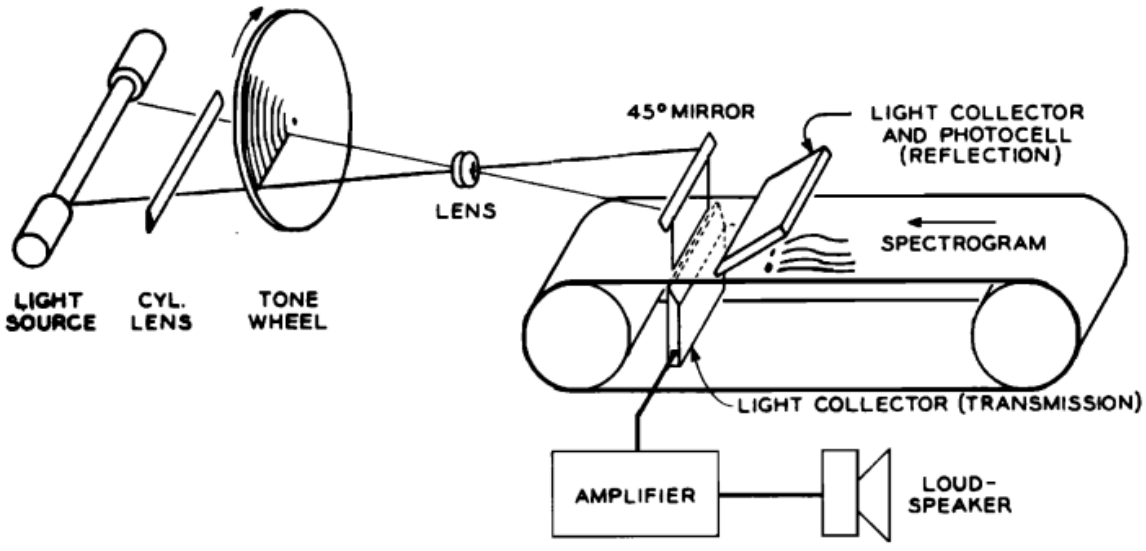


Figure 2.4: SpectrogramReader. Reprinted from Klatt (1987).

Later, In late seventies and early eighties, a tremendous amount research and industry collaborative work concentrated on text-to-speech systems that could generate high quality intelligible and natural speech. Yet, it is in progress.

2.3 Applications of text-to-speech synthesis systems

Text-to-speech synthesis has drawn attention and resources from both researchers and industry. The field has progressed remarkably, and latest text-to-speech synthesis systems no longer sound mechanical and robotic. In the 1980s, due to significant development of technologies in machine learning, digital signal processing, and natural language processing, a more advanced concept of text-to-speech synthesis systems appeared. Recently, progress in the quality aspects of text-to-speech systems such as intelligibility and naturalness of text-to-speech synthesis systems has made it an inescapable component in numerous applications and various domains.

The earliest real-life use of such systems was to assist blind people read books. Even though the quality of early systems was low and robotic, they were widely adopted by blind people in comparison to other available options such as having a real person read the book or reading Braille, mainly because of the availability and usability (Taylor, 2009). Currently, a number of text-to-speech synthesis systems are available in the market to assist the blind to interact with different electronic and smart devices. In the same way, screen readers have existed as an essential tool for people with visual impairment for a long time now. Optical character recognition (OCR) devices and scanners combined with text-to-speech synthesis systems give blind people access to written information and have also assisted them (Dutoit & Stylianou, 2003).

Lately, text-to-speech synthesis has also been commonly utilized by people with difficulties such as reading disorders and also by pre-literate children. Similarly, people with severe speech impairments have widely utilized text-to-speech synthesis through its voice output (Hawley et al., 2013).

Currently, a large number of toys use speech synthesis technologies. For many books, we can turn into audio books which consumers listen to, while commuting to work or in other situations where reading books is not convenient.

Computer-aided learning systems coupled with a high-quality text-to-speech synthesis systems create a supportive environment for learning/teaching a new language.

Text-to-speech synthesis systems are largely employed in entertainment production, especially games and animation. Speech synthesis also provides essential support in diverse research areas such as providing laboratory tools for linguistics and vocal monitoring. Apart from these cases, text-to-speech synthesis systems are involved in reading messages, emails, news, specific-domain related reports, navigation instructions, and a broad range of other applications.

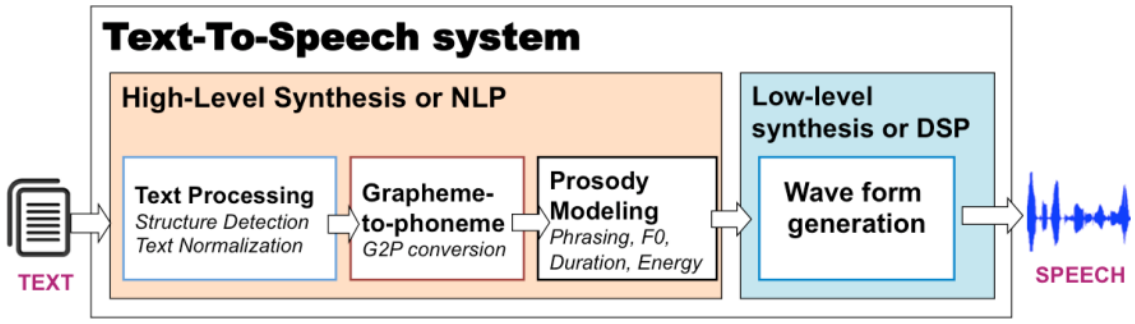


Figure 2.5: Conventional architecture of a text-to-speech synthesis system. Reprinted from Dutoit et al. (2001).

Call center automation is among the important applications text-to-speech synthesis systems, in which textual information can be read over the phone. In those systems, users may pay a bill or book a service and conduct the whole transaction through an automatic dialogue system. In the same way, a substantial application of text-to-speech synthesis is question-answering systems or voice-search, in which users can search for their needed information in a natural input modality and retrieve potentially related information (Taylor, 2009; Dutoit, 1997).

Google Assistant, Apple Siri, Alexa and so on are some typical and well-known voice interactive applications on smartphones whose main component is a multi-lingual text-to-speech synthesis.

These applications allow users accomplish simple operations such as navigation and authentication and more advanced operations such as querying information and processing transactions though a human-like colloquial interface.

2.4 Conventional architecture of text-to-speech synthesis

The conventional architecture of a text-to-speech synthesis system has two main parts , high-level speech synthesis or natural lanuguage processing (NLP) part and low-level or digital signal processing part, with four components: Text Processing, grapheme-to-phoneme (G2p) conversion, prosody modeling, and waveform generation. the first three belong to the natural language processing part and the last one belongs to digital signal processing part.(Huang et al., 2001; Dutoit & Stylianou, 2003). The conventional architecture of text-to-speech synthesis is illustrated in Figure 2.5.

text-to-speech synthesis systems receive raw text as its input. The input text is analyzed and useful phonetic and prosodic features are extracted as follows:

The first component, text Processing deals with the conversion of the input text to a proper format that can be pronounced. The second component, grapheme-to-phoneme (G2p), is

responsible for converting orthographic lexical symbols into a corresponding phonetic sequence. The third component, prosody modeling add necessary prosodic parameters such a pitch and duration to the phonetic sequence. Lastly, the fourth component, low-level speech synthesis, produces the corresponding utterance based on the output of the the third component. We will discuss these four components more in the coming paragraphs.

Text Processing The primary role of this component is to transform all non-orthographic information of text into a script that can be pronounced. Non-orthographic information mainly include special symbols, ordinal and cardinal numbers, different dates formats, abbreviations and acronyms, and other non-orthographic information. Text normalization in this context means changing variety of non-orthographic information into a typical orthographic transcription that is pronounceable. Furthermore, analysis of white-space, punctuation, and other delimiters is essential. These delimiters may have direct implications for prosody such as sentence breaking and paragraph segmentation. Sophisticated syntax and semantics analysis such as obtaining syntactic constituency and semantic features of words , phrases, clause , and sentences are possible, if required, for deeper analysis.

Grapheme-to-phoneme Conversion This component is responsible for converting "lexical orthographic symbols" to a "phonemic representation" along with diacritic information or lexical tones in tonal languages. G2P conversion is easy for languages that have simple and straight forward relationships between phonology and orthography. Spanish is a case in point. Such languages are called phonetic languages. Letter to sound conversion , in its conventional way, is based on rules or dictionary lookup. Pronunciation dictionaries stores the pronounceable form of words. and generates the accurate pronunciation of in text form.

Prosody Modeling prosody modeling is another important in the pipeline. Prosodic features are necessary for the speech to sound natural. Speaking styles moods, and emotions has a huge impact on it. Prosodic features should not only be extracted from the transcribed text but also from the recorded utterance in order to be learned by the model in the training stage. In the synthesis stage, the model should be able to predict these feature from the text and generate natural speech. Modeling it is complicated by the machine learning algorithms have shown the ability to learn prosodic information.

Speech Synthesis This last component, unique to low-level synthesis, receives the input as a sequence of fully tagged phonetics and generates the corresponding waveform.

In essence text-to-speech systems could be developed in one of two ways. Parametric speech synthesis that generates synthetic voices from the parametric representation of speech and concatenative speech synthesis that produces voice by joining mini units of pre-stored human speech. Parametric speech synthesis suffers from a low quality of speech as it is constructed from parameters that are usually produced for a relatively larger units. The second approach suffers while joining the different units of speech. Ideally, the digital signals should be

modified and joined in a smooth and continuous way which could not be the case in many cases. We will touch these approaches in detail later.

Three fundamental classical (apart from deep learning related approaches) techniques in text-to-speech synthesis are:

1. Articulatory text-to-speech synthesis
2. Formant text-to-speech synthesis
3. Concatenative text-to-speech synthesis

The categorization is based on parameterizing text-to-speech synthesis in terms of synthesis and storage.

2.5 Articulatory text-to-speech synthesis

Articulatory synthesis is built entirely upon the physical models of the human speech production system. An articulatory model reconstitutes the vocal tract's shape as a function of the phonatory organs' position. To calculate the signal, the mathematical simulation of air flow through the vocal tract is used. The parameters of the synthesizer are subglottal vocal cord tension and the relative position of the articulatory organs.

The main issue in this strategy is the precise depiction of vocal-track and structure with a limited amount of parameters.

Besides this reason, complexity of human articulation organs limited amount of knowledge are additional reasons (Ipsic & Martincic-Ipsic, 2006) that have contributed to the abandonment of research in articulatory synthesis.

2.6 Formant text-to-speech synthesis

Formant text-to-speech synthesis utilizes the source-filter speech production model, which is based on rules that determine the vocal tract's resonant frequencies. The fundamental frequency must be adjusted and updated at each phoneme which is difficult. Estimation of the vocal tract model is also hard. However, the method can produce quality speech which sounds unnatural.

2.7 Concatenative text-to-speech synthesis

In concatenative text-to-speech synthesis, speech units are stored and linked to produce the speech series of units such as phonemes, diphones, and triphones. The speech generated through concatenative text-to-speech synthesis sounds more natural compared to traditional parameteric techniques.. However, automatic methods/algorithms of dividing waveforms produces audible glitches.

Concatenative text-to-speech synthesis is divided into two subcategories:

1. Diphone concatenation text-to-speech synthesis
2. Corpus-based text-to-speech synthesis

2.7.1 Diphone concatenation text-to-speech synthesis

Diphone concatenation text-to-speech synthesis as a subcategory of concatenative text-to-speech synthesis is based on the same principles. The focal point of this particular subtype is the usage of diphones as concatenative units instead of the phones.

The co-articulation problem blocked the success line of producing high quality speech from the phonemes. Scientist instead tried to adopt a comparatively larger units such as diphones which provide better possibilities to take co-articulation into account. Because the transition from one phoneme to another is smoother compared to larger units, especially when it comes to the joining part. The reason behind is that it is the stable portion of the phoneme and the amount of distortions at the end of diphones are minimum. Hence the amount of smoothing is expected to be minimum. However, The amount of diphones compared to its coressponding phonemes is huge such as about 1500 to 2000 diphones corresponds to jut 40 to 50 phoneme.

In diphone concatenation, linear predictive coding (LPC) is used.Its primary benefits are automatic original signal analysis, simple algorithmic integration, and original sound fidelity. In text-to-speech synthesis, however, LPC was not effective, likely due to its restricted capacity to represent voice parameters

2.7.2 Corpus-based text-to-speech synthesis

Corpus-based speech synthesis is a generalization of concatenative synthesis. It uses dynamic selection of units from a large amount of speech data. The popularity of this method is due to the high-quality synthetic voice it produces.

2.7.3 Preparation of database for corpus based text-to-speech synthesis

Need of an annotated database is the main problem in corpus-based approaches which requiring considerable human effort to mark phonetic boundaries. Several techniques has been used to automate the process. For example, broadband and narrowband edge detection was implemented in Santen & Sproat (1999). Bonafonte et al. (1996) took Gaussian probability density distribution as a similarity measure. In **?**, attempts were made to imitate manual labeling with a set of fuzzy rules using rules-based approach. Mporas et al. (2009) launched a hybrid voice segmentation technique based on hidden Markov models that consists of an iterative isolated phone recognizer unit training which is initialized through embedded training.

2.7.4 Unit selection text-to-speech synthesis

In unit selection synthesis, there are two fundamental methods, although theoretically they are not very distinct. The first technique of unit-selection presented by Hunt & Black (1996) that actually existed before. (Sagisaka et al., 1992). Target cost and concatenation cost plays the vital role. the model will try to minimize the over all cost through optimizing the corresponding function. Target cost indicates that how accurately a candidate unit matches the required unit. It is defined as:

$$C^{(t)}(t_i, u_i) = \sum_{j=1}^{p} w_j^{(t)} C_j^{(t)}(t_i, u_i) \qquad \text{(Equation 2.1)}$$

where u_i represents candidate unit cost and t_i represents the required unit cost and j indexes accross all features(prosodic and phonetic context). The concatenation cost is defined as:

$$C^{(c)}(u_{i-1}, u_i) = \sum_{k=1}^{q} w_k^{(c)} C_k^{(c)}(u_{i-1}, u_i) \qquad \text{(Equation 2.2)}$$

where k, may include spectral and acoustic features. Given the Equation 2.1 and Equation 2.2 costs, optimization is required to find the string units, $u_{1:n} = \{u_1, ..., u_n\}$, from the database and minimize the overall cost of $C(t_{1:n}, u_{1:n})$, as:

$$\hat{u_{1:n}} = \underset{1:n}{\operatorname{argmin}}\{C(t_{1:n}, u_{1:n})\} \qquad \text{(Equation 2.3)}$$

where

$$C(t_{1:n}, u_{1:n}) = \sum_{i=1}^{n} C^{(t)}(t_i, u_i) + \sum_{i=2}^{n} C^{(c)}(u_{i-1}, u_i) \qquad \text{(Equation 2.4)}$$

a general overview of concatenative text-to-speech synthesis is shown in the Figure 2.6

The second way, shown in figure 2.7 uses clustering technique that allows to pre-calculate the target cost effectively (Black & Taylor, 1997; Donovan & Woodland, 1995), then same clusters are mapped into a decision tree which will be queried during the synthesis for available features(e.g., prosodic and phonetic contexts).

There is considerable work on what features to use and how to weigh them. The key to get high quality synthesis in this context is to get the right algorithms, measures and weights.The study of acoustic features and given texts in unit-selection lead to the study of various heuristic or ad-hoc quality measures to form the cost function for target and concatenation costs. Some has proposed statistical models to model cost and concatenation functions as in (Ling & Wang, 2006; Sakai & Shu, 2005; Mizutani, 2002). generally looking into finding weights $(w_j^{(t)}, w_k^{(c)})$ for each feature and the combination of manually tuned and trained weights are used in actual implementations.

A good sized unit is still one of the main issues for selection of of unites in unit-selection speech synthesis. A great deal of research has discussed the various different-sized units, i.e., frame-sized (Ling & Wang, 2006; Hirai & Tenpaku, 2004), HMM state-sized (Donovan & Woodland, 1995; Huang et al., 1996), half-phones (Beutnagel et al., 1999), diphones (Black & Taylor, 1997), to larger unites and even non-uniform units (Segi et al., 2004; Black & Taylor, 1997).

Many parameters can be adjusted and tuned in this context, starting from various sizes of units, various sizes of databases and various scopes of synthesis domains.

To conclude, the concept of "more data" for unit-selection looks as a simple way to follow, but the growth of database to tens of hours of speech, dependence on context and time-dependent differences in voice quality are severe problems (Shi et al., 2002; Kawai & Tsuzaki, 2002; Stylianou, 1999). Likewise, massive databases involve significant computing resources that restrict unit selection methods requiring various voices or various languages.

2.8 Statistical parametric speech synthesis

Statistical parametric text-to-speech synthesis extracts extracts the parameters from the speech such a the fundamental frequency and spectral envelope based on the concept that it could be learned and generated.

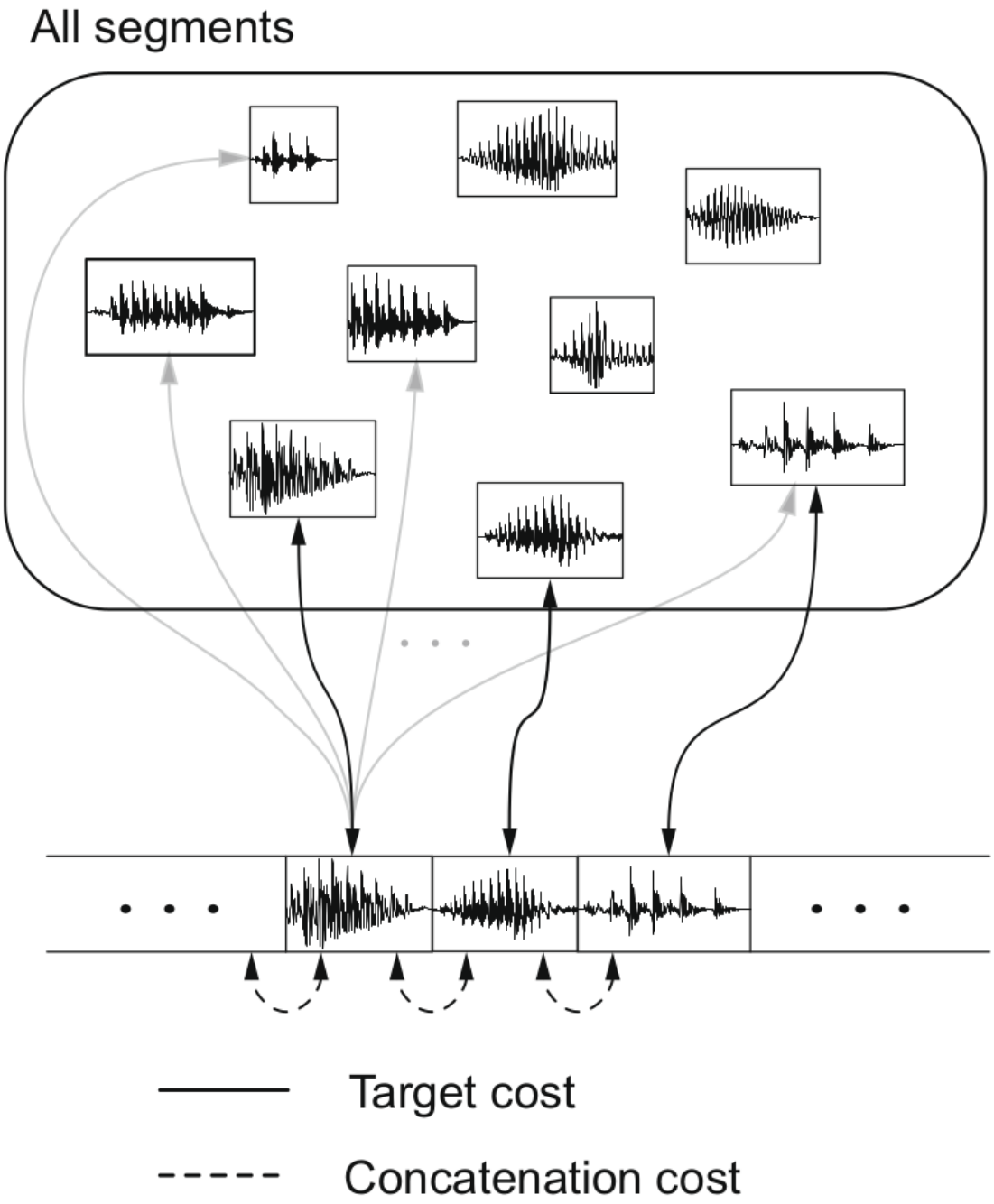


Figure 2.6: Overview of the unit selection scheme with minimization of target costs and concatenation costs. Reprinted from Zen et al. (2009).

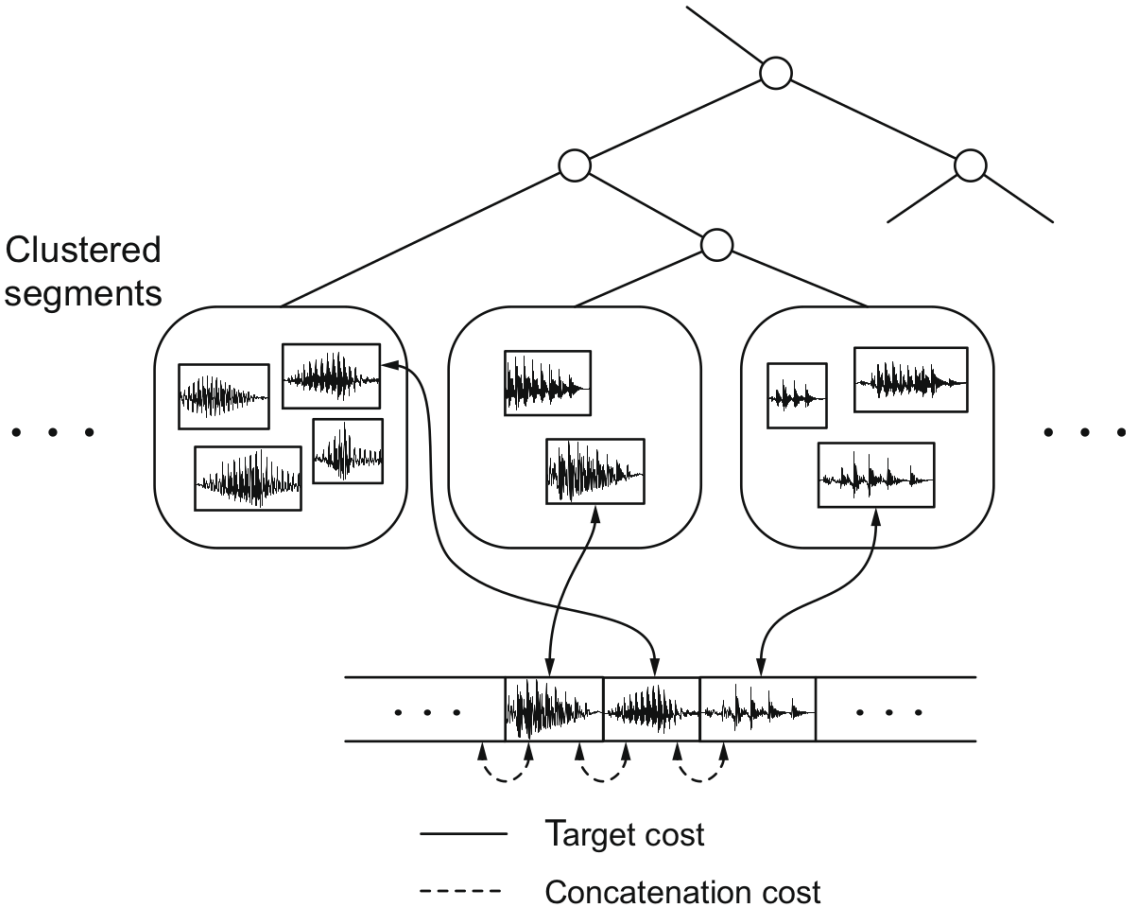


Figure 2.7: Overview of unit-selection scheme. Reprinted from Zen et al. (2009).

The word “parametric” suggests that speech itself is not stored, instead different values are stored which can be reconstructed to create speech.

It considered statistical because of the use of statistical functions (i.e. means, variances and functions of probability density) (King, 2011).

In statistical parametric text-to-speech synthesis, parametric representation of voice from a voice database is extracted . then, a set of generative models such as the maximum likelihood is used to estimate the parameters as:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}}\{p(\boldsymbol{O}|\mathscr{W}, \lambda)\} \qquad \text{(Equation 2.5)}$$

In the equation λ is a set of model parameters, $\boldsymbol{O}$ is a set of training information, and $\mathscr{W}$ is a set of word sequences corresponding to $\boldsymbol{O}$.Next speech parameters are produced, $\hat{\boldsymbol{o}}$, to synthesize a given word sequence, w, set of the estimated models and $\hat{\lambda}$, to maximize the output probabilities as:

$$\hat{\boldsymbol{o}} = \underset{\boldsymbol{o}}{\operatorname{argmax}}\{p(\boldsymbol{o}|w, \hat{\lambda})\} \qquad \text{(Equation 2.6)}$$

Finally, from the parametric representations, speech waveform will be recreated.

2.8.1 HMM-based speech synthesis

Theoretically, any generative model can be used in statistical parametric voice synthesis, but HMMs are most frequently used and researched by a huge number of scientists (Yoshimura et al., 1999). HMM-based model is not an exact representation of actual speech, but effective learning algorithms, automatic model complexity control methods and efficient search algorithms have made HMMs strong models.

2.8.2 HMM-based speech synthesis core architecture

Figure 2.8 show HMM based speech synthesis system which has two separate parts of training and synthesis.

2.8.2.1 HMM Based speech synthesis system training section

In the training part, maximum likelihood estimation of Equation 2.5 based on expectation maximization algorithm (Dempster et al., 1977) is performed. The method as a whole is

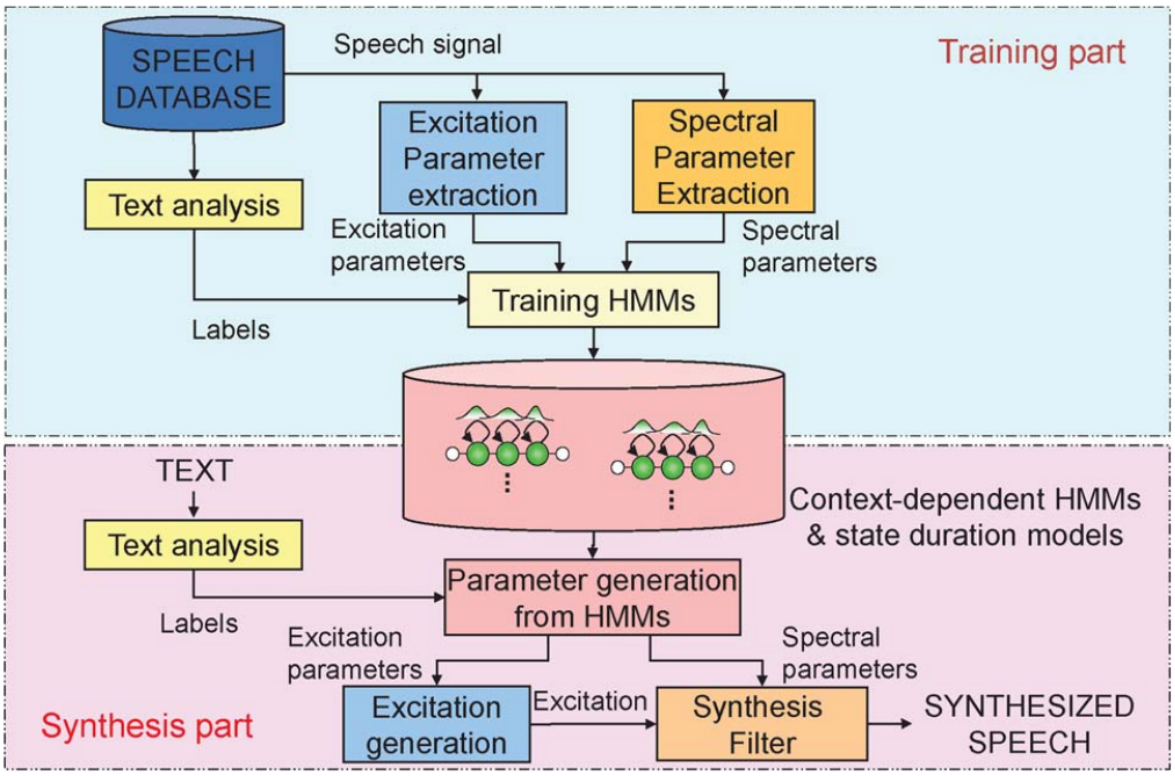


Figure 2.8: Architecture of HMM-based speech synthesis. Reprinted from Tokuda et al. (2013).

comparable to that used in ASR. In the following subtopics, however, there are several variations mentioned.

1. Feature vector and state-output probability:

   Excitation and spectral parameters are extracted as a natural speech database in text to speech and are modeled by a set of multi-stream HMMs dependent on context. Spectral parameters include mel-capstral coefficients (Fukada et al., 1992) and their dynamic features and excitation parameters include $logF0$ and its linguistic features. Since the discrete and continuous distribution both exist in Frames where $F0$ is zero and vice-versa in this case as in figure 2.9. Modeling them with conventional HMMs are not feasible to model $F0$ patterns. several studies has discussed the strategies to overcome the problem (Ross & Ostendorf, 1994; Jensen et al., 1994; Freij & Fallside, 1988). It is chosen to switch between them on the basis of the space label of each observation. Each stream is tailored to synchronize parameters of the spectrum and F0 (Tokuda et al., 2002).

   Next both distributions are stored in variable dimensional observation vector sequences. Multistream HMMs, use distinct distributions of probability of state output to model constituent parts of the observation vector.

2. Explicit duration modeling: The standard HMMs has transition probability model that decreases exponentially as the duration increases. But the model cannot control the

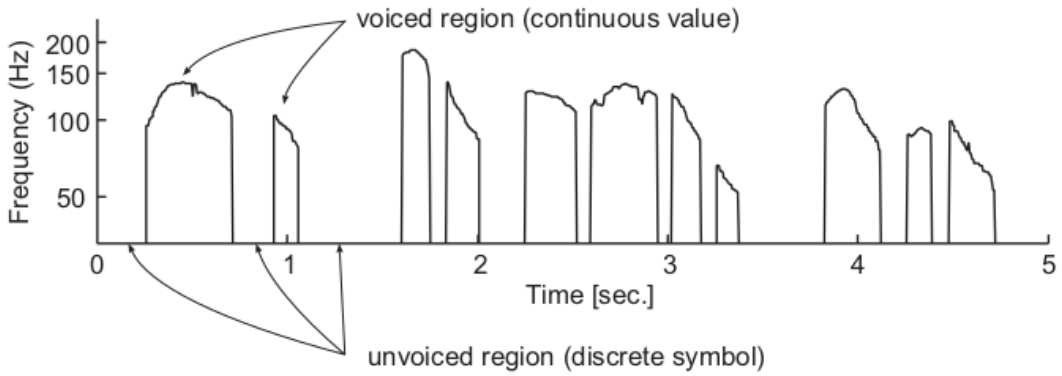


Figure 2.9: Voiced and unvoiced regions of a F0 sequence. Reprinted From Zen et al. (2009).

temporal structure of the speech parameter. Alternatively, HMM-based speech synthesis utilizes Hidden Semi Markov Model (HSMM) where gauussian distribution or gamma distribution approximates the temporal structure (Yoshimura et al., 1998; ISHIMATSU, 2001).

3. Context dependency: In ASR, phoeme/phonetic parameters are mainly used. On the other hand, HMM-based speech synthesis utilizes different linguistic and prosodic parameters for context modeling. Table A.1 illustrates a linguistic and prosodic context dependent label recipe for English language.
4. Parameter tying: This method involves practically too many contextual parameters. To solve the issue, state tying methods are implemented among several context-dependent HMMs to cluster correlated states and tie model parameters.

2.8.2.2 HMM Based speech synthesis systems: synthesis section

Maximization of Equation 2.6 is done in the synthesis. First, convert a series of words to context-dependent sequence. Then, based on the label sequences, construct the utterance HMM by combining context-dependent HMMs.

Second, generate sequences of parameters of spectrum and excitation from the utterance HMMs built in the first phase. several speech generation techniques are used in different studies (Tachiwa & Furui, 1999; Tokuda et al., 2000). Finally, Resynthesize the speech waveform by employing a speech synthesis filter(e.g. mel log spectrum approximation(MLSA) filter (Imai, 1983).

2.8.3 Advantages of statistical parametric text-to-speech synthesis

Flexibility is the key benefit of SPSS. Intelligible speech can be synthesized even with models trained on small amount of data (as little as 100 sentences) because HMM-based speech model is stable and can cover acoustic space despite sparseness of training data. speech model and synthesis engine is small and suitable to use in devices with low computational resources. The speech features, speech styles and feelings can also be changed. And can be easily adapted to a new language.

2.8.4 Drawbacks of statistical parametric speech synthesis

First, the primary drawback of SPSS is lower voice quality. This is primarily due to three variables: vocoder quality, acoustic modeling precision, and over-smoothing in the process (Zen et al., 2009).. Since mel-capstral vocoder is used with simple periodic pulse-train or white noise excitation, the synthetic speech sounds buzzy. High-quality vocoders need to be integrated to reduce the issue . Secondly, HMMs are effective in describing transitions between states but regarding a specific state the parameters are static. State-output probability depends on the current state and the probability factor for duration decreases exponentially that is not the situation with genuine voice . To overcome the problem, extra models to describe the duration should be deployed. The statistical average makes it possible for the model to generate smooth trajectories but results in muffled sounding and because the trajectories produced by the voice parameter are over-smoothed, the natural variability of speech is discarded.

2.8.5 Differences between speech synthesis of unit selection and HMM

Statistical parametric speech synthesis (SPSS) is an effective solution for overcoming the limitations of unit selection. SPSS utilizes statistical machine learning methods like HMMs to deduce the parameter-mapping specification from data. In the most simple form it can be defined as producing the average of some sets of likewise sounding voice fragments. On the other hand, unit-selection synthesis as a subtype and natural extension of concatenative speech retains natural and unmodified speech units and is focused on the issues such as handling huge amount of units, prosody be extension beyond fundamental frequency and time control, and how to how to reduce signal processing distortion (Taylor, 2009).

While both SPSS and unit selection depends on data, in SPSS general properties of the data is learned whereas in unit-selection the data is effectivly memorized. Similarly, both techniques use features of speech units but the usage is different. Unit selection utilizes both contextual characteristics (phonetic and prosodic context) and speech characteristics (spectral and acoustic characteristics) to minimize the target cost (the finest units) and the concatenation cost (the finest sequence) of units for an arbitrary utterance, Whereas, in its training and synthesis stages, HMMs-based synthesis utilizes contextual characteristics (phonetic, lin-

guistic, and prosodic). Contextual characteristics are forced to align with voice parameters while training to create context-dependent HMMs. In the synthesis stage, the sequence of context-dependent labels is created based on the contextual characteristics and finally utterance HMMs are created by their concatenation.

Both of these techniques have received considerable amount of attention and resources from academia and industry to enhance the quality of synthesized speech but both have pros and cons as some of them are discussed as follows:

Unit-selection speech synthesis is more suitable for applications with limited/closed domains such as announcement systems (train stations, airports, and so on) and call centers($24/7$ services). In contrast, HMMs speech synthesis is more appropriate for open domains such as SMS, emails, news, question/answering systems, and speech translation. Based on Zen et al. (2009)'s review, the results of Blizzard challenge in 2005 and 2006 mean opinion scores and word error rates statistics has shown that HMMs based speech synthesis was more recommended and understandable. However, the finest examples of speech synthesis for unit selection were better than the previous ones. From the computational perspective, speech synthesis based on HMMs requires less memory to store model parameters (duration and acoustic model statistics), It therefore needs less runtime than unit-selection text-to-speech synthesis as it is storing multiple versions of the similar units. One of the major drawbacks of unit-selection based speech synthesis in comparison to HMMs based speech synthesis is its severely downgraded quality of speech if the sentence required necessitates contexts (phonetic context or prosodic context)that are not represented or under represented in the repository. Although it may be deemed a scarce incident, a small poor join in an utterance can ruin the flow of the listener. Thus, poor and/or improper joins can not be avoided because the number of possible combinations is large and cannot be covered even with a huge database. However, HMM's voice synthesis is more robust to noise or fluctuations due to recording circumstances or shortage of some voice units (Zen et al., 2009). Unit-selection based speech synthesis require lots of parameters tuning by hand in HMMs base speech synthesis is based on mathematically well defined statistical parameters.

The table 2.1 summarized the major differences between unit-selection and HMM based speech synthesis.

2.9 Deep learning components in text-to-speech synthesis systems

The use of deep learning in text-to-speech synthesis is very recent and currently the state of the art belongs to a particular set of architectures in it. To avoid confusion, lets keep the following points in mind. First, deep learning can be interpreted as another variation of statistical parametric speech synthesis, because neural networks can mimic the underlying statistical model. To maintain constant notion, HMM-based synthesis is referred as SPSS and deep learning based synthesis as DNNs synthesis.

The early works of deep learning uses feed forward neural networks as acoustic models for

Criteria	Unit-selection synthesis	HMM-based synthesis
Approach	Data-driven: memorize data (natural speech)	Parameter-driven: learn properties of data
	Multi-template	Statistics
Idea	Retain natural unmodified units by selecting appropriate sub-words	Generate the average of some sets of similarly sounding segments
Preferred applications	Limited domain	Open domain
Techniques	Target cost, concatenation cost	Machine learning
	Single tree	Multiple trees (spectral, F0, duration)
Quality	Discontinuity at the join	Smooth
	High quality at waveform level	Vocoded speech (buzzy)
	Less preferred	More understandable
	Best examples are better	Best examples are worse
Footprint	Large run-time data	Small run-time data
Robustness	Hit or miss (with spurious errors, quality is severely degraded)	Stable
Voice modification	Extremely difficult	Flexible to change speaking types voice characteristics or emotion
	Fixed voice	Various voices

Table 2.1: Summary of differences between HMM-based speech synthesis and unit-selection concatenative speech synthesis. Based on (Zen et al., 2009)'s work.

SPSS. For instance, Ze et al. (2013) uses DNN in the acoustic mapping process. In SPSS the acoustic mapping is the generation of acoustic parameters out of the Gaussian mean from the proper cluster. In this case, DNNs emit the predictions which substitute the decision tree with the Gaussian distribution. The application of DNNs obtained better modeling of complex context dependencies and outperformed the classic approach of decision trees.

Similarly, Lu et al. (2013) combines a vector-space representation of linguistic context with DNNs which adapts contiguous representation directly to make acoustic mapping. Qian et al. (2014) employs DNNs to map acoustics from linguistic inputs. Another essential point in their work is the study of various parameters that affects convergence schemes of text-to-speech synthesis with moderate sized corpus. Their scheme outperformed the conventional HMMs and the main improvement came from the prosody prediction (concretely with F0 contour).

Hu et al. (2015) uses dynamic sinusoidal model (DSM) (Hu et al., 2014) in a DNN-based acoustic model with multitask learning. First they model cepstra for spectral parametrization and then the log-amplitudes as a direct parametrization of the DSM. In the synthesis, they fused the second task with the first one and got improvement in performance.

Kang et al. (2013) approaches the problem using Deep Belief Network (DBN) generative model to show the dependencies between linguistic and acoustic characteristics. The experiment demonstrated win against classical SPSS approach.

Zen & Senior (2014) claim that using DNNs to model the acoustic mapping has some limitations. They proposed deep mixture density network (MDN) to address two of them. First, DNNs cannot model complex distributions except unimodel Gaussian distributions. The second limitation is that the output of DNNs provides mean values only, while the variance is an important property in achieving naturalness. Their MDN provides a set of outputs that model gaussian mixture models of each output acoustic parameter along with its means and variances. Their findings show that this architectural shift can relax the constraints of acoustic modeling based on Deep neural network.

Wu, Valentini-Botinhao, et al. (2015) addresses two problems in application of DNNs to the speech synthesis: perceptual sub-optimality and frame-by-frame independence. They evaluated that the first problem is due to the training criterion and the second one is due to the difficulty of optimizing recurrent neural networks and its computational expensiveness. Perceptual sub-optimality arises because the training focuses on maximizing the probability of acoustic characteristics that represent poor perception of human speech. In addition, expected perceptual error does not reflect the error in speech features accurately. To solve the problem they proposed a multi-task learning (MTL) procedure, in addition to predicting the typical invertible vocoder parameters as the main task, DNNs learn to predict a perceptual representation of the target speech as a secondary task. The MTLs serves as hints during the main task training and will be discarded during the the synthesis. Regarding the second problem, they proposed a simpler technique called bottleneck feature stacking. In this technique, a DNN with bottleneck hidden layer is trained first and then the activations of the bottleneck which yield a compact representation of both acoustic and linguistic information for each frame of many contiguous frames are stacked together, joined with the linguistic features and passed into another DNN stage where acoustic maps are made.

Similarly, Wu & King (2015) uses stacked bottleneck activations to have a wide context with a training criterion that minimizes the trajectory errors by taking dynamic constraints from a wide acoustic context. This leads to minimized utterance-level trajectory error instead of frame-by-frame error and better naturalness compared to the previous model.

We can see quite big amount of work on speaker adaptation and capturing speakers voice characteristic based on DNNs. For example,Wu, Swietojanski, et al. (2015) carried out DNN speaker adaptation with three types of techniques: adding identity information to the input features, Learning hidden unit contribution (LHUC)(Swietojanski & Renals, 2014), and making output feature space transformations.

In Fan et al. (2015) proposed a model to hold many speakers out of the same shared DNN structure with a specific training mechanism where they back-propagate all the speakers information in the same mini-batch. The process achieved better results with the multitask approach compared to learning a single speaker parameters in isolated manner. Then transfer the learning of the base shared structure for a new speaker to achieve speaker adaptation with limited training data. Good results in naturalness and similarity to the original speaker were achieved.

On the other hand, RNNs and its variants (Hochreiter & Schmidhuber, 1997) have influenced the sequence processing and prediction, which leads to interesting results in the speech syn-

thesis.

S.-H. Chen et al. (1998) explored the usage of standard RNN architectures with many hidden layers synchronized by different timings (at syllable level and at word level) for prosodic parameters prediction such as contours of syllable pitch, concentrations of syllable energy, initial and final syllable duration, and duration of inter-syllable pauses. Achanta et al. (2015) investigated two variants of RNNs applied to acoustic parameter generation: Elman-RNN and Clockwork-RNN. They demonstrated that Elman-RNN is equal to Clockwork-RNN with a specific type of Leaky Integration (LI) (Bengio et al., 2013).

Even though, The LSTM is the most commonly used RNN for speech processing apps. Fernandez et al. (2014) used a bidirectional LSTM architecture to predict F0 contours. Zen & Sak (2015) employed a unidirectional LSTM architecture to make a low-latency speech generation model. unidirectional LSTM with recurrent output layer achieves better results compared dynamic parameters predictions and deriving the trajectory using Tokudas algorithm (Tokuda et al., 2000).

Wu & King (2016) focused on the effectiveness of LSTMs. Especially on the necessity of the different gating mechanisms and they come up with a more efficient solution that only requires forget gate and input gate. The interesting point is that the forget gate is the inverse of input gate and thus needs not to be learned. Lastly, Coto-Jiménez & Goddard-Close (2016) proposed a post-filtering methodology to be performed for LSTM in SPSS. The goal is to enhance the performance of predicted speech in terms of closeness to natural voice compared to HMM synthesis.

2.10 End-to-End deep learning speech synthesis models

We saw in the earlier sections that the text-to-speech synthesis systems have multiple stages and components such as text analysis component to extract different linguistic characteristics, a duration model, an acoustic prediction model and a complicated signal processing vocoder (Agiomyrgiannakis, 2015; Zen et al., 2009) are common. Wang et al. (2017) mentions the following main restrictions in a typical statistical parametric text-to-speech synthesis pipeline:

1. Each component in the pipeline requires extensive expertise in the field and design is laborious..
2. Each component is trained separately, so that mistakes can accumulate in each component.
3. Building a new system is complex and requires to substantial amount of engineering efforts.

On the other hand, end-to-end text-to-speech synthesis with minimal human annotation can be trained on ■اود، تشت pairs which alleviates the laborious feature engineering process.

Secondly, It enables conditioning of different characteristics such as speaker, language, or high-level characteristics such as sentiments. This is due to the fact that the conditions can occur in the beginning of the model. Thirdly, It might also be easier to adapt new data. Fourthly, It is more probable that single models will be robust than multi-stage models, and finally, this model will enable us to train a considerable amount of rich and expressive data. yet often noisy data.

and even in DNN based speech synthesis. DNNs were not used to map the whole process. but were used instead of some components such as regression trees. In this section, we are reviewing the works where the authors claim that their deep learning modules can be trained from the raw input <text, audio> pairs. Thus eliminating the sophisticated, individual-independent modules and processes, which may often contained brittle design choices, from the pipeline. We will discuss WaveNet, Tacotron, and DeepVoice.

2.10.1 WaveNet

WaveNet is influenced by latest developments in the generation of neural auto regressive models that model complex distributions such as text (Jozefowicz et al., 2016) and images (A. van den Oord, Kalchbrenner, & Kavukcuoglu, 2016; A. van den Oord, Kalchbrenner, Espeholt, et al., 2016). It is based on PixelCNN's architecture. The authors believe that WaveNet is a generic and flexible solution not only for text-to-speech synthesis, but also for all other audio generation apps such as music, speaker transformation, source segregation and speech enrichment.

WaveNet works on raw audio waveform directly. As a product of conditional probabilities, the joint probability of waveform $\mathbf{x} = \{x_1, \cdots, x_T\}$ is factorized as follows:

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t | x_1, \cdots, x_{t-1}) \qquad \text{(Equation 2.7)}$$

The model produces a categorical distribution with a softmax layer over the next x_t value. Each audio sample is therefore conditioned on all prior time-steps

The main ingredients of WaveNet are causual convolutions which is equivalent of masked convolutions for images (A. van den Oord, Kalchbrenner, & Kavukcuoglu, 2016). Causal convolutions, particularly when applied to very lengthy sequences, are trained faster than RNNs.

In order to improve the receptive field, causal convolution requires many layers or large filters. Therefore a slightly different convolution called dilated convolution is used. The idea is similar to pooling or strided convolution except that the output has the same size as the input in this case. Dilated convolutions have previously been used in the various

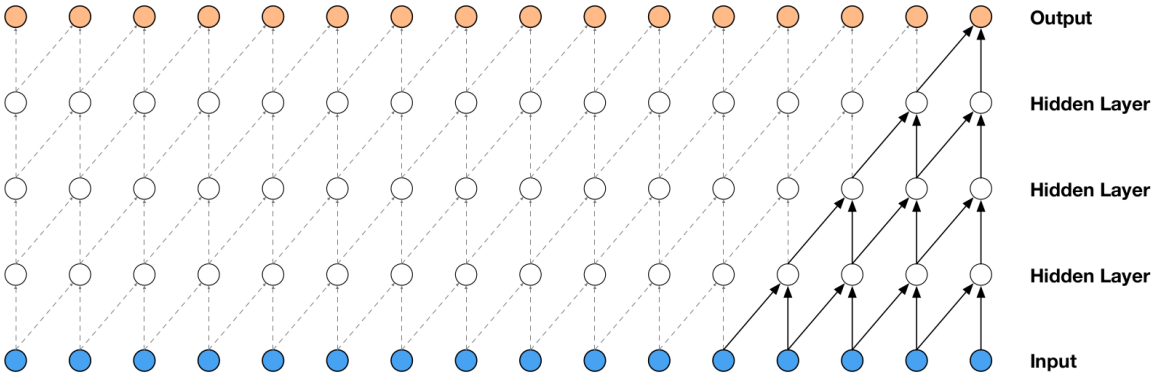


Figure 2.10: Stack of causual convolutional layers. Reprinted from Van Den Oord et al. (2016).

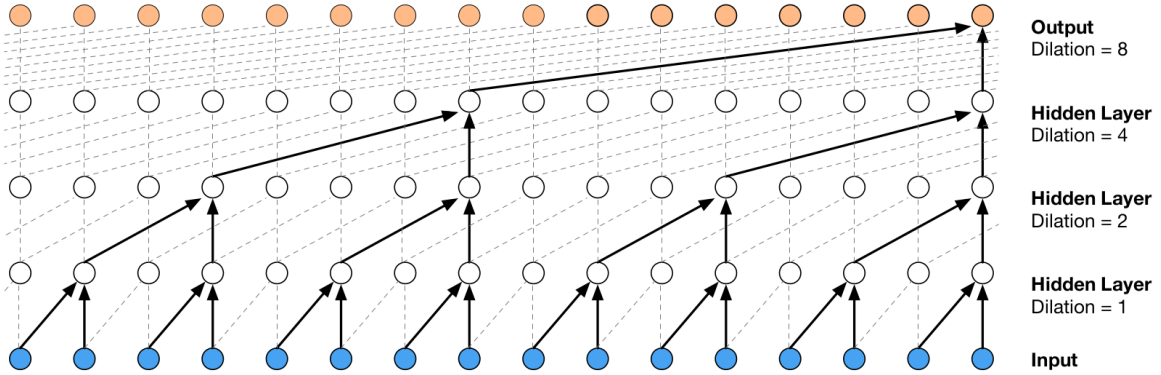


Figure 2.11: Stack of dilated causual convolutional layers. Reprinted from Van Den Oord et al. (2016).

contexts such as image segmentation (L. C. Chen et al., 2014; Yu & Koltun, 2015) and signal processing (Holschneider et al., 1990; Dutilleux, 1990).

WaveNet uses stacked dilated convolutions to enable very large receptive fields, preserve the input throughout the network, and maintain computational efficiency. as shown in the Figure 2.11

Another interesting point is the architecture of the gated activation units where element-wise multiplication of sigmoid and tanh is used to compute the activation function.

$$\mathbf{z} = tanh(W_{f,k} * \mathbf{x}) \odot \sigma(W_{g,k} * \mathbf{x}) \qquad \text{(Equation 2.8)}$$

g denotes the gate, $\odot$ denotes an element-wise multiplication, * denotes a convolution operator, $\sigma(\cdot)$ is a sigmoid function, k is the layer index, f denotes filter, g denotes the gate, and W is a learn-able convolution filter.

Moreover, residual and parametertized skip connections are used to speech up the convergence and help the training of much deeper models. Figure 2.12 shows the architecture of gated activation unites, residual, and skip connections.

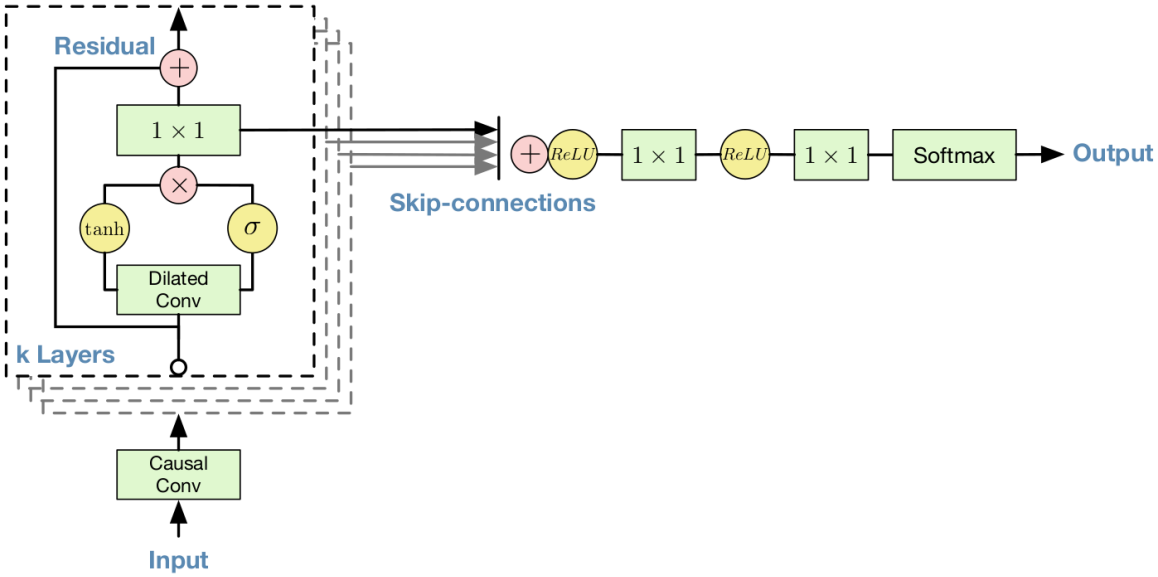


Figure 2.12: Overview of the gated activation units, residual blocks, skip connections, and the entire architecture. Reprinted from Van Den Oord et al. (2016).

In the text-to-speech related experiments, they used 24.6 English dataset.

For the evaluation, subjective paired comparison tests and mean opinions score (MOS) tests are conducted. the Table 2.2 shows the summary of the tests.

Table 2.2: Subjective 5-scale mean opinion scores of previous models and WaveNet. WaveNet significantly enhanced the past state of the art, Reducing the difference between natural speech and best prior model by more than 50 percent. Reprinted from Van Den Oord et al. (2016).

	Subjective 5-scale MOS in naturalness	
Speech samples	North American English	Mandarin Chinese
LSTM-RNN parametric	3.67 ± 0.098	3.79 ± 0.084
HMM-driven concatenative	3.86 ± 0.137	3.47 ± 0.108
WaveNet (L+F)	$\mathbf{4.21} \pm 0.081$	$\mathbf{4.08} \pm 0.085$
Natural (8-bit μ-law)	4.46 ± 0.067	4.25 ± 0.082
Natural (16-bit linear PCM)	4.55 ± 0.075	4.21 ± 0.071

Despite the interesting results, WaveNet has some shortcomings. First, it is not a truely end-to-end system yet. It requires text-extracted linguistic features such as phone identities, syllable stress, and others as its front-end. Secondly, It is computationally expensive. To

overcome such challenges various other models such as Fast wavenet (Paine et al., 2016), Parallel WaveNet (A. v. d. Oord et al., 2017) , and Clarinet (Ping et al., 2018) has been proposed to overcome the computational expensiveness. Similarly, some works tried some variations of of WaveNet such as tacotron 2 to make the inference process faster and developed a complete end-to-end architecture to overcome the first limitation.

2.10.2 Tacotron

Tacotron is another generative model. Given the <text, audio> pairs, Tacotron takes a a sequence of text as input and outputs its spetrogram. It is based on sequence to sequence learning with neural network (Sutskever et al., 2014) with attention paradigm (Bahdanau et al., 2014) inspired by neural machine translation.

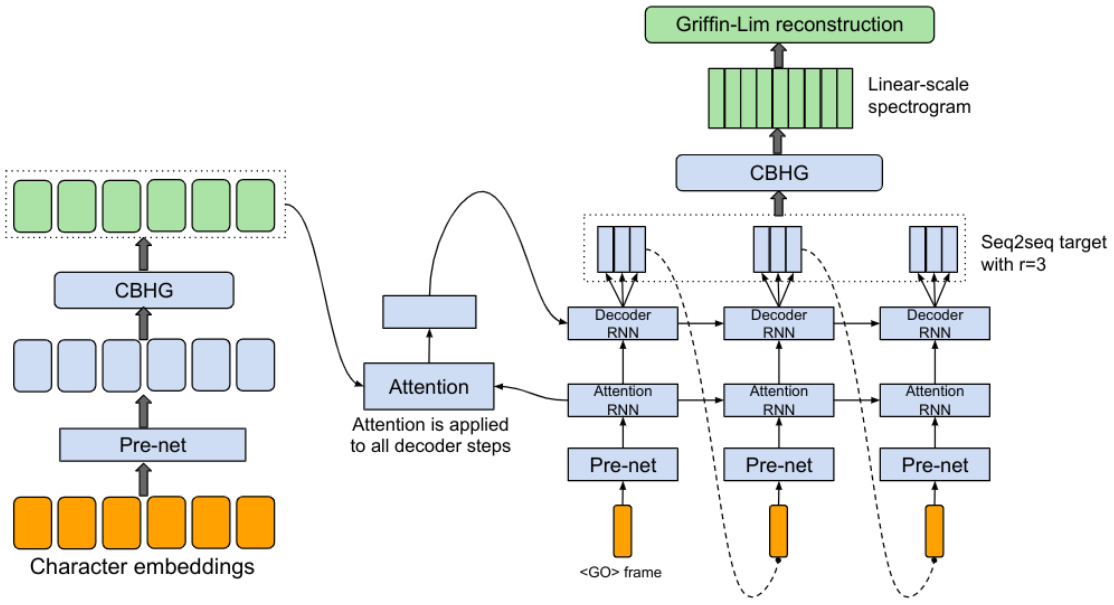


Figure 2.13: General architecuture of Tacotron 1. Reprinted from Wang et al. (2017)

Figure 2.13 shows the overall architecture of the Tacotron that is made of four modules: The CBHG (1-D convolution bank + highway network + bidirectional GRU), encoder, decoder, and a post-processing net and waveform synthesizer modules. The details of architecture and is hyperparameters are shown in table 2.3.

The base architecture of vanilla sequence to sequence has been improved by many techniques as shown in the 2.3. It frame based which makes the inference faster compared to sample-level auto-regressive methods.

For model training, an internal North American English dataset which contained about 24.6 hours of speech data

Table 2.3: Architecture and is hyper-parameters of Tacotron 1. Reprinted from Wang et al. (2017)

Spectral analysis	*pre-emphasis*: 0.97; *frame length*: 50 ms; *frame shift*: 12.5 ms; *window type*: Hann
Character embedding	256-D
Encoder CBHG	*Conv1D bank*: K=16, conv-k-128-ReLU *Max pooling*: stride=1, width=2 *Conv1D projections*: conv-3-128-ReLU $\rightarrow$ conv-3-128-Linear *Highway net*: 4 layers of FC-128-ReLU *Bidirectional GRU*: 128 cells
Encoder pre-net	FC-256-ReLU $\rightarrow$ Dropout(0.5) $\rightarrow$ FC-128-ReLU $\rightarrow$ Dropout(0.5)
Decoder pre-net	FC-256-ReLU $\rightarrow$ Dropout(0.5)$\rightarrow$ FC-128-ReLU $\rightarrow$ Dropout(0.5)
Decoder RNN	2-layer residual GRU (256 cells)
Attention RNN	1-layer GRU (256 cells)
Post-processing net CBHG	*Conv1D bank*: K=8, conv-k-128-ReLU *Max pooling*: stride=1, width=2 *Conv1D projections*: conv-3-256-ReLU $\rightarrow$ conv-3-80-Linear *Highway net*: 4 layers of FC-128-ReLU *Bidirectional GRU*: 128 cells
Reduction factor (r)	2

As it is hard to compare generative models based on objective metrics, Tacotron relies on visual comparisons and mean opinion score (MOS) . Table 2.4 shows 5-scale mean opinion score evaluation for Tacotron.

Table 2.4: Taçotron MOS

	Mean opinion score (MOS)
Tacotron	3.82 $\pm$0.085
Parametric	3.69 $\pm$0.109
Concatenative	4.09 $\pm$0.119

2.10.3 Deep voice

Deep voice 1 is developed by Baidu. It is influenced by traditional text-to-speech synthesis pipeline and adapted a similar structure. Deep voice 1 is composed two procedures: the training procedure and the inference procedure. and five major components: grapheme to phoneme, segmentation, phoneme duration, fundamental frequency, and an audio synthesis model shown in the Figure 2.14.

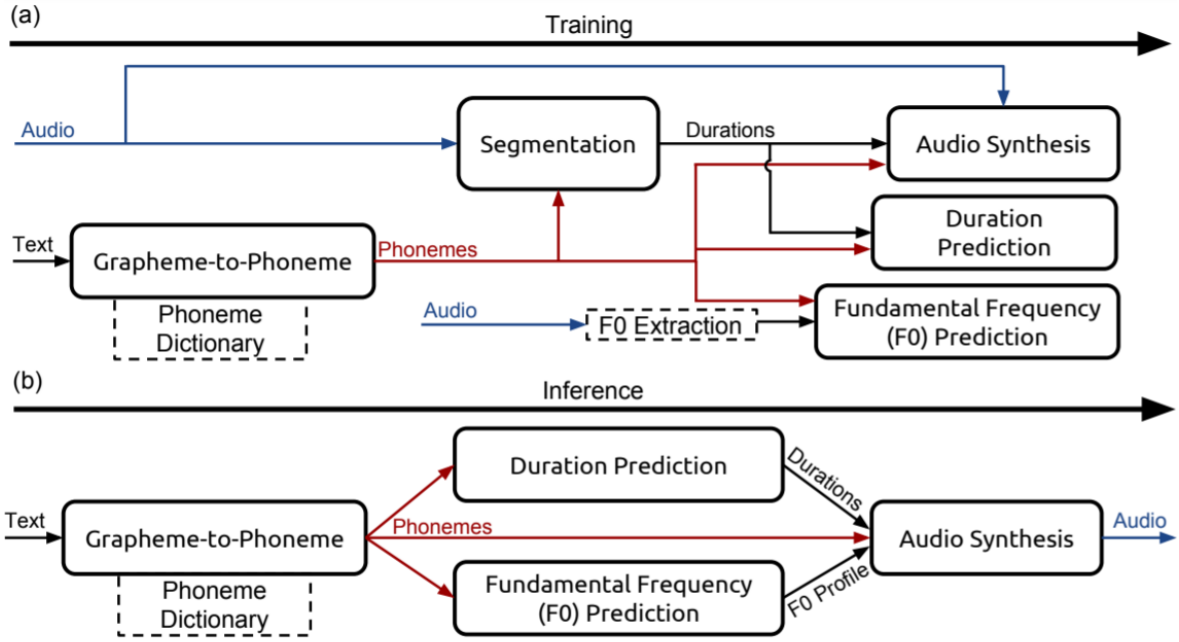


Figure 2.14: System diagram depicting training and inference modules, with inputs on the left and outputs on the right. In our system, the duration prediction model and the F0 prediction model are performed by a single neural network trained with a joint loss. The grapheme-to-phoneme model is used as a fallback for words that are not present in a phoneme dictionary, such as CMUDict. Dotted lines denote non-learned components

The first model transforms the written text into phonemes based on an encoder and decoder. The segmentation model locates the phoneme boundaries for precise detection and is trained to predict phoneme pairs rather than single phonemes. The featurization is carried out through the computation of 20 MFCCs with milli-second stride. Given an utterance and the phoneme by phoneme transcription of it, the model will predict the beginning and the end of the phoneme in the time-dependent audio. The phoneme duration and the fundamental frequency model estimate three feature for each phoneme with a single model. The predicted features include: the phoneme duration, the probability of the either the phoneme is voiced or not, and 20 time-dependent fundamental frequency values. The model of audio synthesis combines the grapheme result with the model of phoneme duration and the fundamental frequency to synthesize the audio at a high sample rate. The audio synthesizer is a variant of of WaveNet (Van Den Oord et al., 2016). However, It is 400x faster compared to the orignal WaveNet. The author have provided separate analysis section that shows a glimpse of performance and limitation for every section.

A 20.5 hours from 9741 utterances subset of Blizzard 2013 dataset is used for experiments. For the evaluation, 5-Likert scale mean opinion score (MOS) is used and shown in the Table 2.5

There are other variations of end-to-end model such as deep voice 2 (Gibiansky et al., 2017),

Table 2.5: Mean Opinion Scores (MOS) for utterances.

Type	**Model Size**	MOS±CI
Ground Truth (48 kHz)	None	4.75 ± 0.12
Ground Truth	None	4.45 ± 0.16
Ground Truth (companded and expanded)	None	4.34 ± 0.18
Synthesized	$\ell = 40, r = 64, s = 256$	3.94 ± 0.26
Synthesized (48 kHz)	$\ell = 40, r = 64, s = 256$	3.84 ± 0.24
Synthesized (Synthesized F0)	$\ell = 40, r = 64, s = 256$	2.76 ± 0.31
Synthesized (Synthesized Duration and F0)	$\ell = 40, r = 64, s = 256$	2.00 ± 0.23
Synthesized (2X real-time inference)	$\ell = 20, r = 32, s = 128$	2.74 ± 0.32
Synthesized (1X real-time inference)	$\ell = 20, r = 64, s = 128$	3.35 ± 0.31

deep voice 3 (Ping et al., 2017), and Tacotron 2 (Shen et al., 2018) emerged one after another outperforming the previous ones. Most of these models are based on the previous versions, only replacing some components of them with some others. Tacotron 2 is among the latest ones that has outperformed all the previous models and reported the mean opinion score (MOS) of 4.526 ± 0.066 with 95% confidence intervals computed from t-distributions.

Chapter 3

Methodology

This chapter provides a detailed overview of the process I use to build a TTS system for the Pashto language. I illustrate the step-by-step procedures of corpus creation, DNN-based model developement, input feature extraction and engineering, output feature extraction and engineering, and evaluation of the overall system.

3.1 Methodology Overview

As discussed in Chapter 1, this work involves the following steps.

- Corpus creation
- Two-stage DNN model development
  - Front end analysis
  - Input feature extraction and engineering
  - Output feature extraction and engineering
  - Speech synthesis
- Testing intelligibility and naturalness of the synthesized text

The proposed model is depicted in Figure 3.1.

3.2 Corpus Creation

In machine learning, dataset/corpus preparation is the first step. Sometimes there are pre-existing data that a we can make use of directly. The data scientist will analyze the patterns in the data, heuristically dive deeper, and try to make sense of the data in new and efficient ways.

Unfortunately, there is no corpus available online for the Pashto language. Thus, I prepared the corpus manually. The preparation has two stages: extracting and normalizing script and recording the speech.

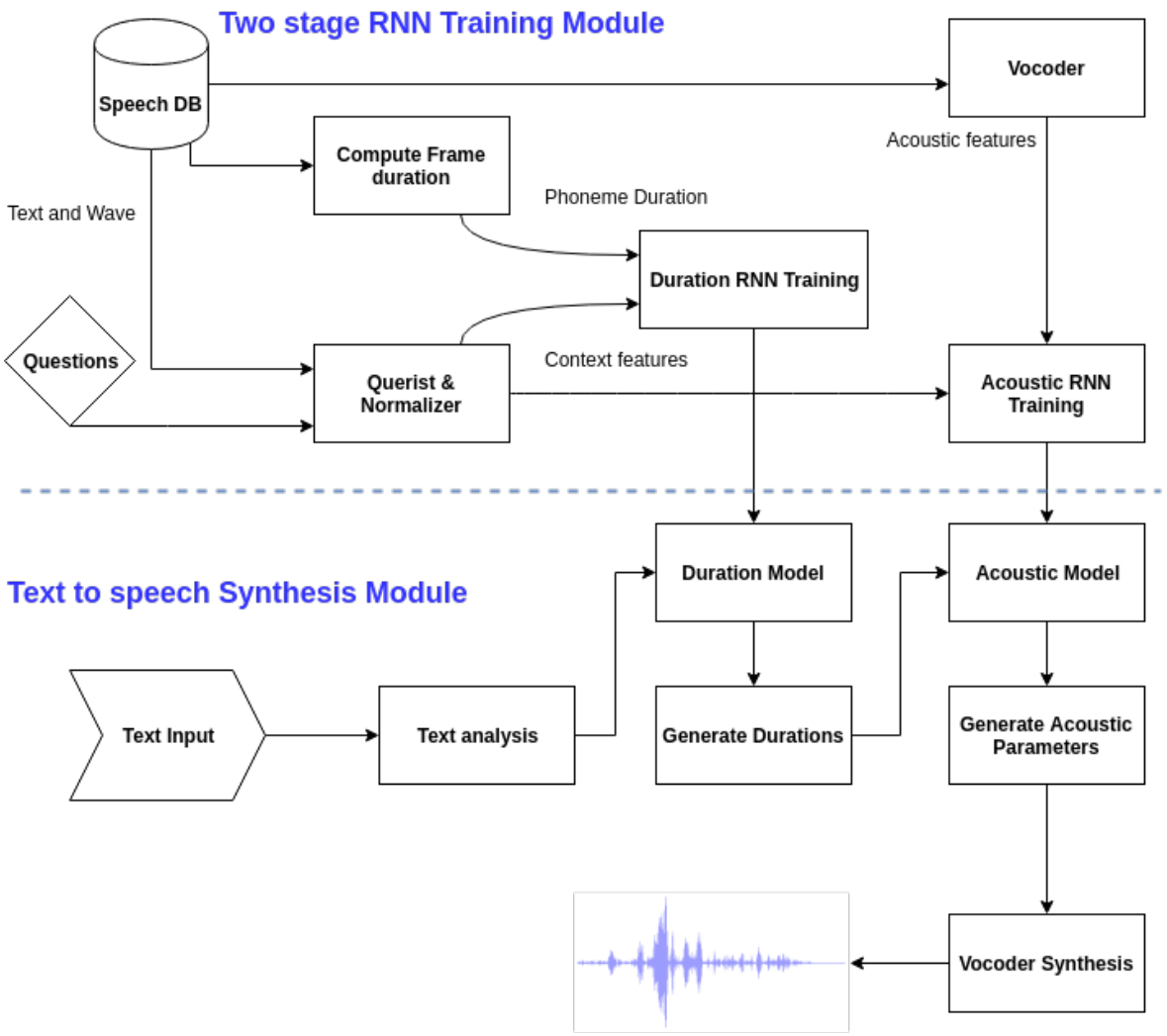


Figure 3.1: Two stage model with training and synthesis modules. Based on the work of Wu et al. (2016).

3.2.1 Extracting and normalizing script

Since this is the first Pashto language corpus to be used in text to speech, I will mainly focus on preparing the dataset with minimal mismatch between the script and corresponding utterances. At the same time, I minimize the normalization problem during the preparation phase of scripts manually to reduce the burden of preprocessing activities for my research work and for further research. This will allow researchers to focus on improvement of TTS and ASR systems for the Pashto language.

One of the main obstacles that has limited research on the Pastho language, especially in TTS, ASR, and machine translation, is the absence of datasets. I make the dataset freely accessible online to be used for research purposes. I hope that this dataset will pave the way for more research in TTS and ASR for the Pashto language.

3.2.2 Recording the speech

During the evaluation of different speech databases for the English language, I found that most data were produced by professional speakers, trained to master clear pronunciation, stress, and intonations in professional studios. Those datasets have almost perfect conditions in terms of recording quality and noise reduction. Unfortunately, I am neither a professional speaker, nor do I have access to a studio. However, I will enhance the quality as much as possible by adhering to standard procedures such as using a professional microphone to record the speech and recording when the surrounding noise is minimal.

Regarding dialects, my pronunciation is similar to the pronunciation of people living in the eastern, central, and northern regions of Afghanistan. I ignore deviations of my own dialect and stick with standard and formal pronunciation.

3.3 Data preparation

3.3.1 Front end analysis

Analysis and labeling is carried out through a conventional front end. Generally, it includes the following components: tokenization, part of speech tagging, letter-to-sound (LTS) rules, phrase breaks, and intonations. The combination of the above components will output linguistic specifications. All of the those components have to be learned individually from labelled data or compiled manually based on an expert's knowledge. This conventional approach to building a front end is expensive in terms of time, data collection efforts, and expertise.

To bypass the above tasks, open source toolkits such as Festival, MaryTTS, and eSpeack are available. Those tools provide all the needed features for some languages.

No tools are currently available for Pashto yet. Developing each of those components requires a tremendous amount of work and expertise. In this work, I use Ossian. Although it does not have all the feature that Festival or others provide, it can provide me with essential feature such as tokenization, distributional word vectors as POS tag substitutes, letters as substitutes for phonemes, forced alignment and silence detection, and phrasing.

3.3.2 Normalization

Text often include highly context dependent abbreviations and acronyms. Technical reports may include formulas, figures, charts, graphs, tables and its corresponding captions. Data from users interaction among themselves may require interpretation of symbols such a emoti-

cons. Emails may have web addresses and other special formal or informal abbreviations such as “FYI” which means for your information or “IMHO” which means in my humble opinion. Again, any text document may include part numbers, ordinal numbers, cardinal numbers, account numbers, dates, times, money and currency, mathematical expressions, chemical formulas, etc, that needs to be changed into a single canonical and directly pronounceable/spoken form. The method of creating standardized orthography from the text that includes various representations is text normalization. It is an essential step in text to speech synthesis. However, in speech to text or dictation systems requires the inverse process of text normalization which is more challenging than the normalization process itself to tackle.

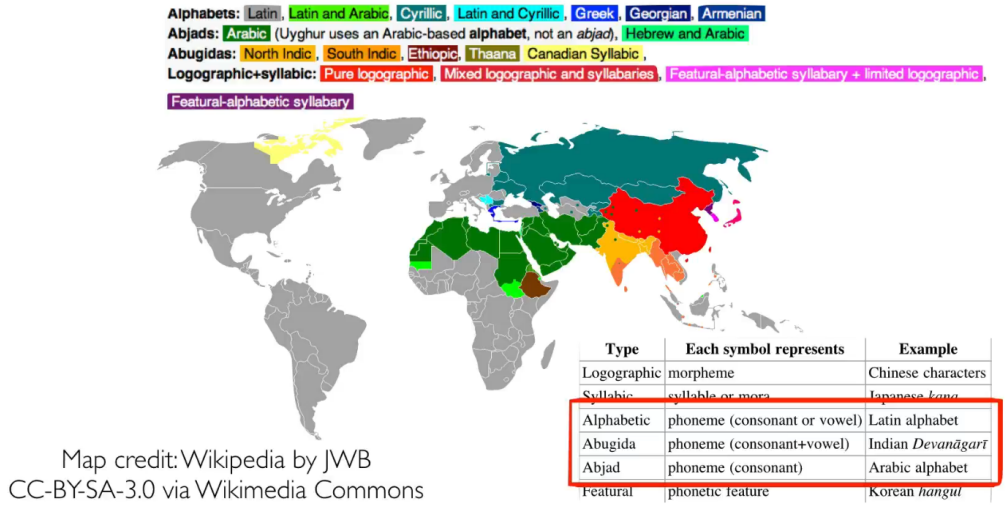
3.3.3 Tokenization

Tokenization is one of the common pre-processing tasks in natural language processing. Given a set of texts and chopping it up into pieces called tokens is referred as tokenization. Tokens are loosly referred to as words or terms, but sometimes its important to make the distinction between them. Coming up with a tokenizer to cover all the cases of a specific language is highly dependent on the peculiarities of that language and writing a precise tokenizer to work for all the languages is not practical. However, based on syntax similarity of some languages, some simple and general techniques could be developed that would cover most, if not all, of the cases in similar languages (left to right or right to left with spacing between words)

3.3.4 Letter to sound

To understand the letter to sound concept from scratch, lets dig into the process of writing as a form of communication and find the category that Pashto belongs to in different writing system. In conveying massages, both writing and speech are useful but writing differs in being a reliable form of information transfer and storage. A writing system is a shared and understandable method of representing visual information between writers and readers. Writing systems can be divided into broad groups such as alphabets, syllabaries, and logographies. A particular system may have inherited attributes from more than one the above systems. Figure **??** shows different writing systems on the map.

In the alphabetic group, vowels and consonants are represented by different symbols or graphemes. Abjads, which Pashto belongs to, are a little different from classical alphabetic group. It is consonant-based and do not have separate symbols for vowels and consonants. Next one is abugidas. It is consonant-based too, but It uses diacritics to denote vowels. Syllabaries and logographics are on the other extreme. In syllabaries, each letter represents a syllable or mora. Finally, in logographies each sign correlates to a word, morpheme, or other semantic units. The total number of symbols in the last two groups to represent a full language is huge and may reach to hundreds of symbols/signs. refs(Wikipedia)



Type	Each symbol represents	Example
Logographic	morpheme	Chinese characters
Syllabic	syllable or mora	Japanese *kana*
Alphabetic	phoneme (consonant or vowel)	Latin alphabet
Abugida	phoneme (consonant+vowel)	Indian *Devanāgarī*
Abjad	phoneme (consonant)	Arabic alphabet
Featural	phonetic feature	Korean *hangul*

Figure 3.2: Several writing systems such as alphabets, abjads, abujidas, etc, are shown along with the geographical locations that they are used prominently .

In conventional approach to text to speech synthesis especially in English, phonemes are generally used to instead of raw letters. CMU Pronouncing Dictionary (CMUdict) is one of the most common open-source machine-readable pronunciation dictionary for North American English created by the Speech Group at Carnegie Mellon University (CMU) for research in speech recognition. CMUdict mainly provides orthographic/phonetic for English words in their pronunciations which is used to generate representations for speech synthesis and speech recognition. The recent version 0.7b contains over 134000 words and is actively maintained and expanded. In CMUdict, 39 phonemes are used to code the English transcription (lexical stress variations excluded). Vowels may also carry three extra lexical stresses For instance, CMUdict transcribes “Homework” as “/HH OW M W ER K /” without stress and as “/HH OW1 M W ER2 K/” with lexical stress. ref(CMUsite)

3.3.5 Forced alignment and silence detection

For the forced alignment, HTS force alignment toolkit is used to extract the timing information such as silence in the utterance, the start of a letter, the end of a letter, and sub-phone information from the the utterance and then the extracted information is appended into the front-end features file.

3.3.6 Phrase breaks

A naive but effective way for phrase break detection in languages where a separate component for phrase breaks are not available is using silences as a proxy for prosodic phrase breaks. The use of silence as a proxy where information from both natural language and

speech is combined helps a lot and the need for a separate phrase identifier could be bypassed. Although it is not highly accurate but given a little filtering with minimum silence duration can provide satisfactory results.

3.3.7 Input feature extraction and engineering

Input feature extraction and engineering refers to analysis and labeling of raw text input and preparing it to be passed as input to a DNN model. The main steps in this phase are obtaining and flattening linguistic specifications, attaching contextual information to phones, encoding each context-dependent phone as a vector, encoding most using binary features, upsampling using duration information, changing duration into the frame sequence, and adding fine-grained positional information to the sequence.

3.4 Output(acoustic) feature extraction and engineering

Waveforms as an output of a speech systems is a vital part. Its analysis, manipulation, and synthesis is still among the important topics in research.

Since the utterance is the result of a text to speech system. The features should be extracted from the speech signal in a way suitable for modeling that can be used to reconstruct the waveform. The problem with speech signals in this domain is that the variation is huge even for a phoneme. For example the Figure 3.3 shows two different variations for a /a:/ phoneme.

Based on the observation, both of the signals represent the same phoneme but can not be seen easily as it is high variant. If both of the signals are of a similar or the same sound, there must exist some techniques to extract the pattern and show that the signals are of the similar sounds. For example, If the spectrum is computed using a regular fast Fourier transform (FFT) a more meaningful behaviour or pattern could be seen intuitively as shown in Figure 3.4 but the variation is still high and cannot be seen even intuitively with a subtraction of two waveforms.

One of the possible solutions, used by many researcher, is to extract spectral envelope as shown in the Figure 3.5. It shows very similar signals that are near to the ground truth that both of of the signals represent the same /a:/ phoneme. There are some attempts of using the raw waveforms but it has disadvantages such as too much extra information in the signal.

3.4.1 Spectral envelope estimation

A very high level overview of spectral envelope estimation using cheap trick algorithm (Morise, 2015) includes the following steps: designing a window function based on the

• Phoneme /a:/

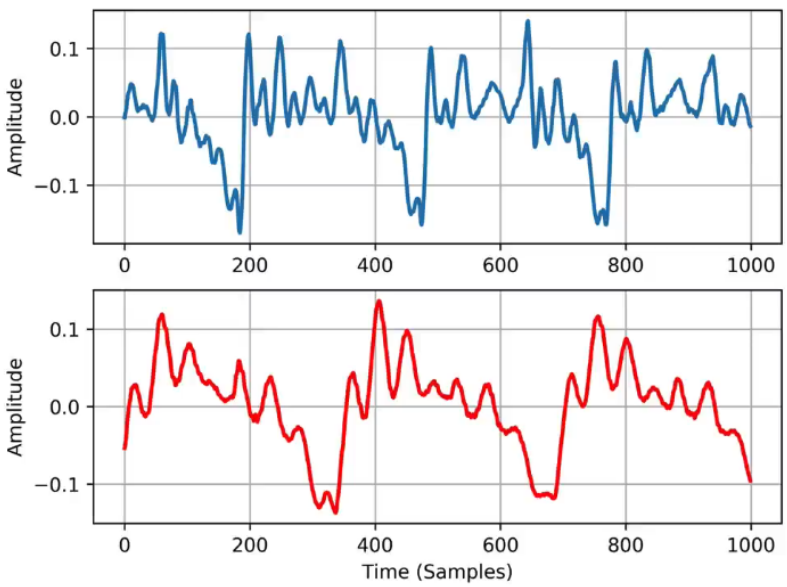


Figure 3.3: Two different signals for the phoneme /a:/ are shown. The signals does not look to be of the same phoneme intuitively but can easily be recognized that both have the same sound when heard by human.

idea of pitch synchronous analysis (Mathews et al., 1961) with a Hanning window of length of 3 pitch period ($3T_0$) is used. as shown in Figure 3.6.

To illustrate the whole process, consider the speech signal shown in the Figure 3.7 where the first step of Hanning window of length of 3 pitch period is applied. The result of other steps will be plotted on the same signal for consistency and illustration purposes. The second step is smoothing of the power spectrum by applying a moving average filter of size $(2/3)F0$ to get rid of the spikes that goes to negative numbers which makes the signal more stable as shown in Figure 3.8.

Afther the first smoothing process, another smoothing process of $2F0$ is applied to that makes the curve look like a spectral envelope as Shown in Figure 3.9. Still the curve needs to lifted up, Finally, a weighted sum of shifted version of the given curve is applied taken as spectral envelope of the waveform as shown in Figure 4.6.

3.4.2 Fundamental frequency estimation

To extract the fundamental frequency (F0) using DIO, an interval detection method that consists three steps is applied as shown in the Figure 3.11.

The first step in the process is low-pass filtering with different cut-off frequencies. The second step is the calculation of the variances of negative and positive going zero-crossing

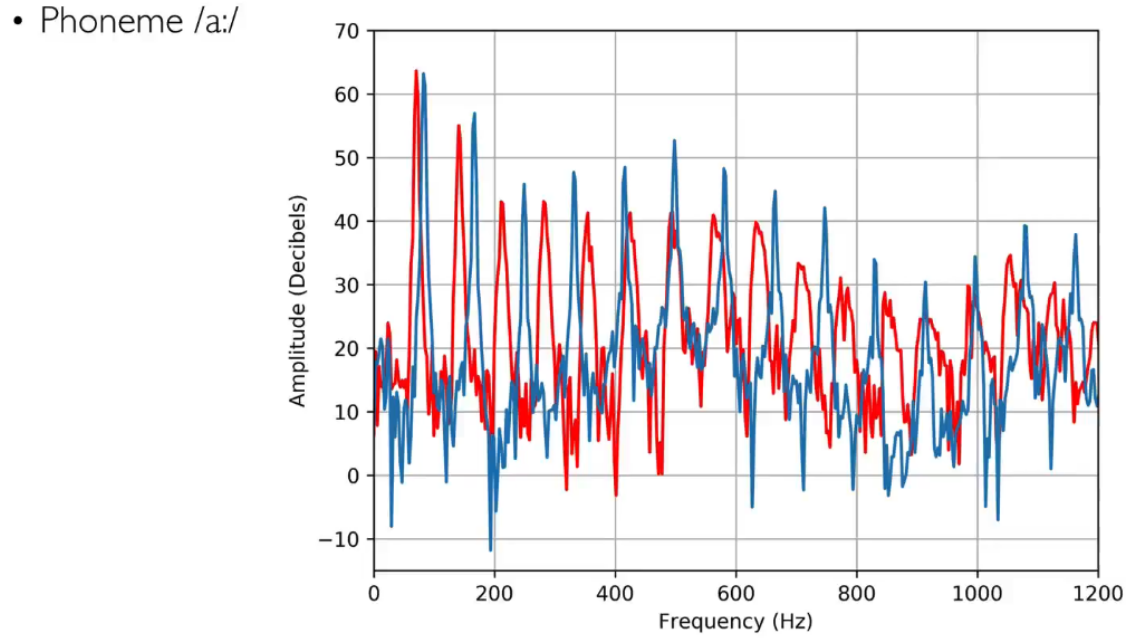


Figure 3.4: The result of the fast Fourier transform applied on both signals is plotted.

intervals and the intervals between the successive peaks and dips (The Fundamentalness) and the final step the selection of lowest fundamental frequency (F0) based on the fundamentalness from all the candidates as the final fundamental frequency (F0). Figure 4.7 depicts the variation of fundamental frequency (F0) extracted using DIO algorithm.

3.4.3 Band aperiodities estimation

The final important acoustic feature is band aperiodicity or in simple words, the degreee of randomness in a certain band. It is calculated as the ratio between aperiodic and periodic energy, averaged over certain frequency bands. i.e. it is calculated in D4C by the division of "total power" over the "sine wave power". In the Figure 3.13 three bands: lowest in band as "**a**", more in band as "**b**", and highest in band as "**c**" are shown. band aperiodicity shows that if the a signal is more random and has less harmonics the band aperiodicity value will be bigger and vice-versa.

To this point, the extraction process of three important acoustic feature is explained. The results of the process are raw vocoder features from the waveforms which are usually high in dimensions. i.e. the dimensions of extracted spectral envelope are $fft\_length/2$ which may be around 1000 to 2000 coefficients or more. an extra step to process the extracted information in such a way that is convenient in terms of number of coefficients and format to be passed to neural network is required. This process is called output feature engineering.

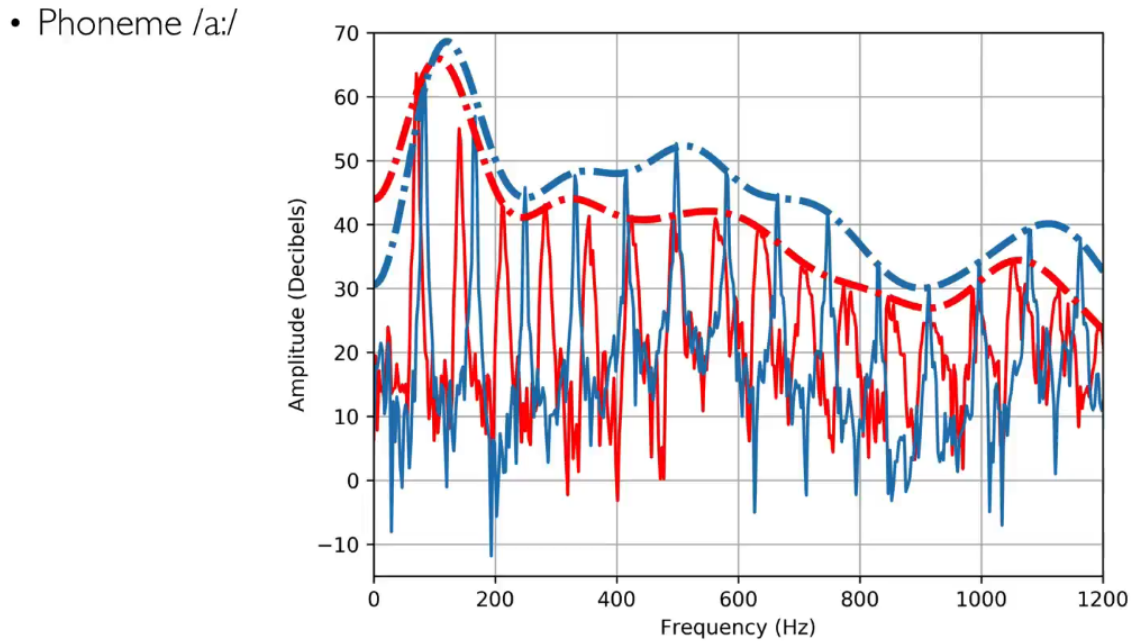


Figure 3.5: Spectral envelope of both signals of /a:/ phoneme are extracted and plotted on the same window. The pattern of similarity could easily be seen between the red and blue spectrum. This similarity depicts that both signal are representing a similar sound.

3.4.4 Output(acoustic) feature engineering

Figure 3.14 depicts the overall process of acoustic feature engineering. The process originally developed for old parametric speech synthesis. Many aspects of this process could be analyzed differently and based on research changed in a better way.

Lets stick to the order from the prevoius section and start from spectral envelope as our first step. If the spectral domain is transformed to cepstrum domain and only 60 coefficients are kept, most of the information could be maintained and at the same time thousands of parameters are compressed to 60 parameters only. The second step is to take the $logF0$ and then interpolate. The reason main reason is that the out put of the neural network is Gaussian/Unimodel but the values for F0 as can be seen intuitively in Figure 4.7 are not Gaussian as most of the values are in range of $90 - 140Hz$ and similarly around $0Hz$. To enforce Gaussianity, first $logF0$ is taken and then the unvoiced frames are interpolated (zero frames are removed). The problem with taking $logF0$ and interpolation is the information loss as the information of unvoiced frames (zero frames) are lost. To overcome the information loss problem in the process an extra bit of information called voicing decision is appended to the features. It is a binary value indicating voiced/unvoiced frames. The third step is to normalize the features by mean and variance to maintain ratio and make it efficient to be processed by neural networks. Finally, appending the delta, and delta-delta dynamic features. These features indicate how do these features evolve overtime.

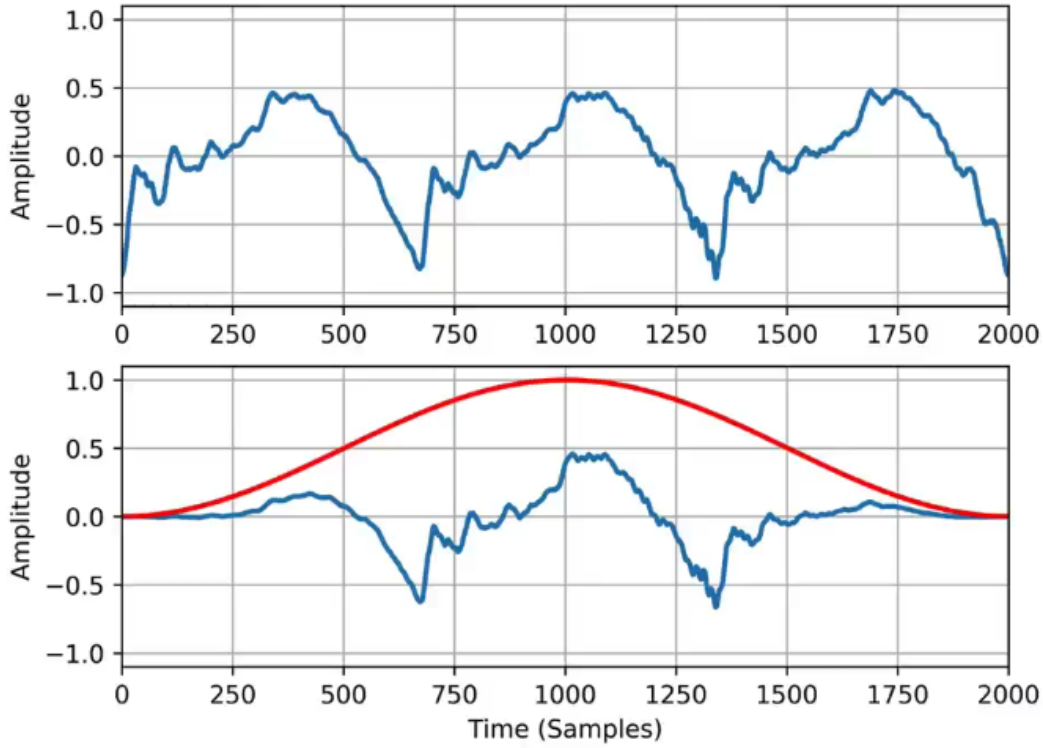


Figure 3.6: The first step of cheaptrick algorithm is shown. The first plot shows a speech signal and the second plot shows a Hanning window of length of 3 pitch period applied on the same speech signal.

3.5 Two stage RNN-LSTM Training Module

Based on the proposed methodology, the RNN-LSTM scenario is implemented as shown in Figure 3.1. Two models, namely duration model and acoustic model are trained. Both of the models could be trained with different architectures and various parameters but based on time and space constraints only one of those models is discussed in detail. However, a holistic evaluation of some other models and experiments will be discussed briefly in the evaluation. Furthermore, explanation of RNN and LSTM architectures and how they work is skipped. the interested reader can find it in most of the deep learning related books.

As seen in 4.3 and 4.4, two types of features are extracted. These features are required to produce quality voice. The first type of information (linguistic features/input features) helps in understanding the context such as duration, intonation, pauses between sentences, phrases and words, etc, are referred to as prosodic information. Prosodic information has a huge effect on the naturalness of the generated speech.

The second type of information (acoustic features/output features) are the mathematical rep-

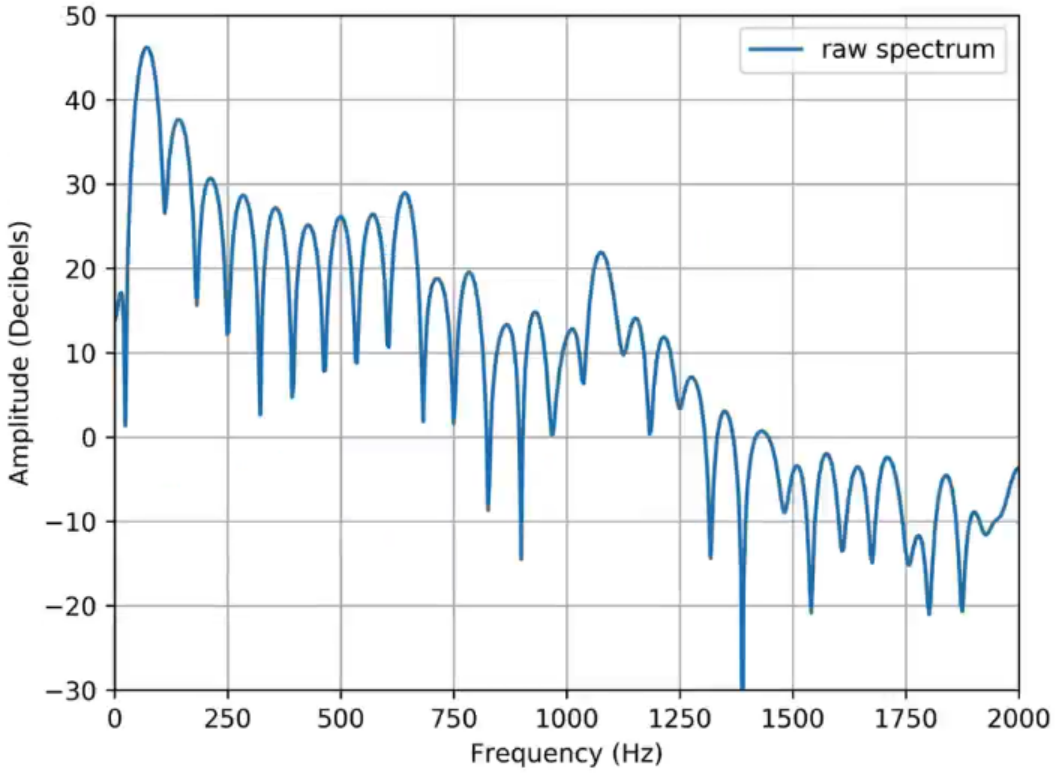


Figure 3.7: Raw spectrum extracted with Hanning window of length of 3 pitch period is shown as a result of the first step

resentation of the speech such as: spectral estimations, extraction of fundamental frequency $F0$, and estimation of different bands of periodic and aperiodic energy. Only with a good estimation of mentioned parameters, intelligible and also near to natural speech could be produced.

The prosodic or more specifically the phoneme duration prediction problem is approached first and a duration model/predictor is trained to decide the amount of frames to be produced for a given text-input.

The next phase is the prediction of acoustic parameters for given amount of frame in a continuous manner. This is where the logic of two stage RNN-LSTM comes into the scenario:

- To predict the duration from the linguistic features, duration model is build as the first stage of the module development.
- To predict the acoustic features for as many frame as predicted by the duration model, acoustic model is built as shown in the Figure 3.1.

The recurrent neural network and its variants are well-suited to this task because of the following characteristics:

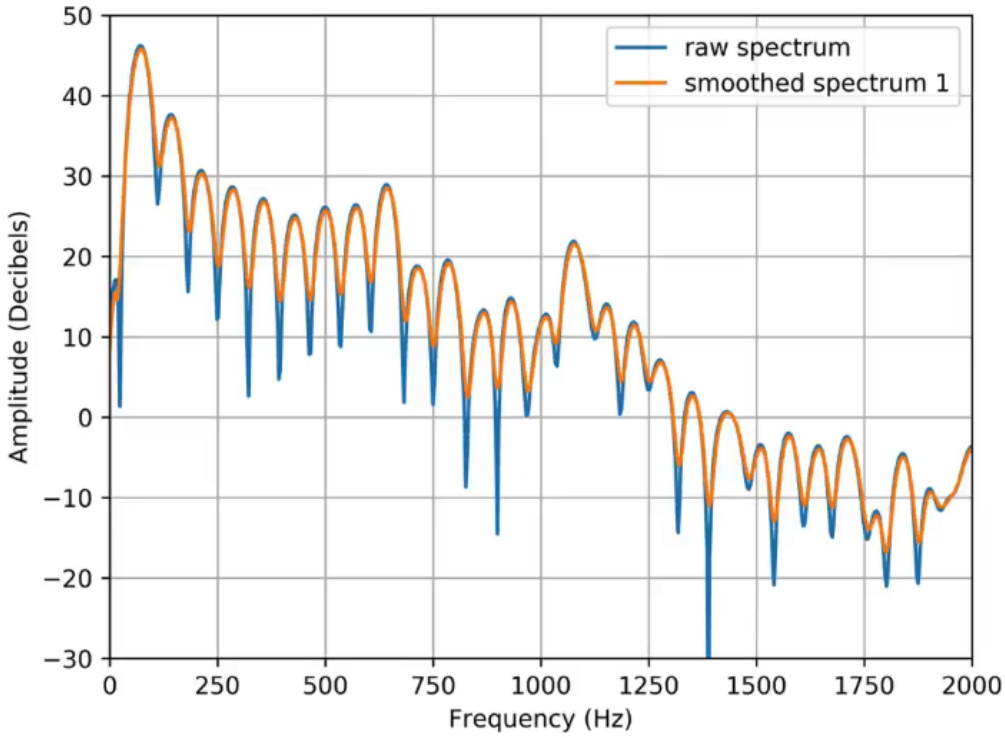


Figure 3.8: The orange color spectrum shows a comparatively smoother signal after applying a moving filter of size $(2/3)F0$ to get rid of the spikes that goes beyond zero to negative numbers

- We keep track of sequential context while reading input in a timely manner, one phoneme after another. Thus, each step encodes a label.
- The RNN's output layer improves the continuity of the predictions between frames in the acoustic prediction stage (Zen & Sak, 2015).

The DNN architecture for both of the models is RNN. The duration model is purely RNN and the acoustic model is RNN-LSTM. In both of the implementation "tanh" activation function is used. for complete set of parameters refer to Table A.2 and **??**.

The choice of RNN and RNN-LSTM is based on the recent works in speech synthesis for English language. The main advantages of using RNNs among all others are:

1. RNNs in its generic form emerged to solve problems where tracking the sequential context is necessary as it is the case with speech synthesis where a long input is read in a timely fashion (state by state/phoneme by phoneme) and the amount of frames calculated.
2. Having RNN output layers improves the continuous prediction of acoustic parameters (Zen & Sak, 2015) far more better than the previous SPSS techniques.

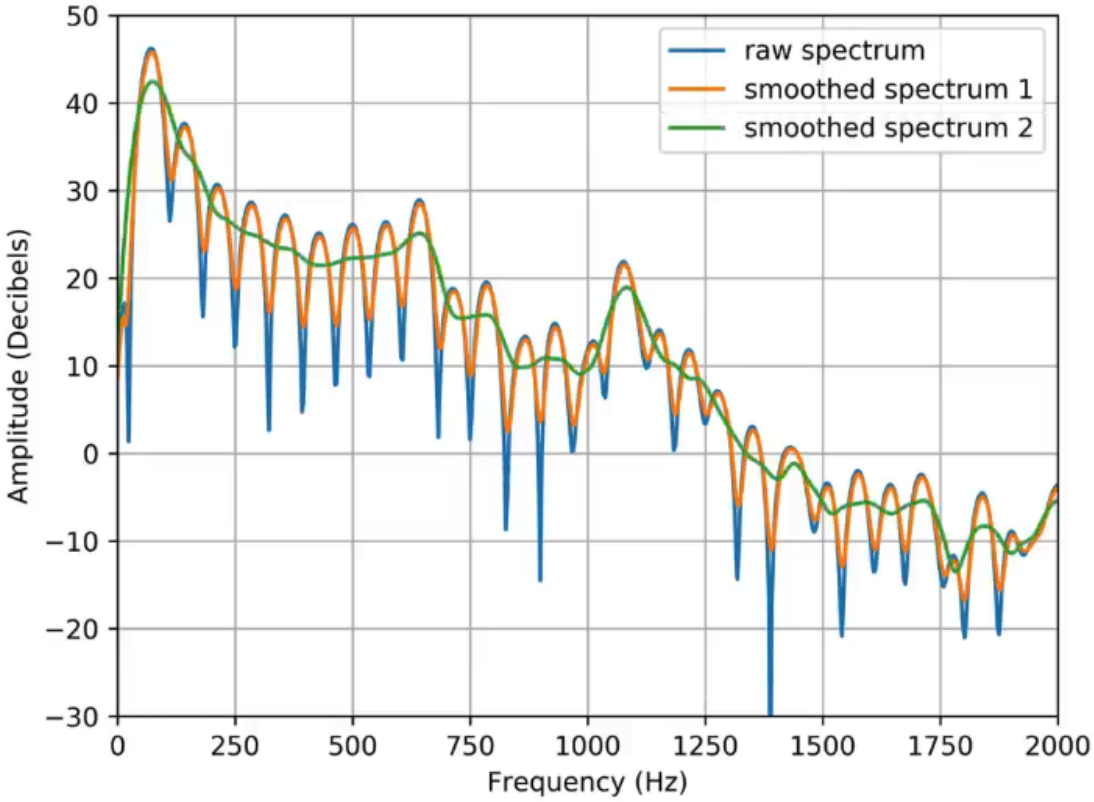


Figure 3.9: The green color spectrum shows a more smoother signal after the second smoothing process is applied. It looks less variant and more like spectral envelope.

The proposed two-stage RNN training module for Pashto text to speech is shown in Figure 3.15.

3.6 Speech synthesis module

To produce synthesized speech for a new text input, it has to be analyzed and prepared in a format that can be passed to a pre-trained duration model to generate duration, and the duration is passed to the the pre-trained acoustic model to generate acoustic parameters. Finally, the acoustic parameters have to be passed to the vocoder to produce the utterance of the input text. The process is shown in Figure 3.16.

3.7 Evaluation of accuracy, intelligibility, and naturalness of the synthesized text

Evaluation of the whole TTS system is not trivial, due to the involvement of many components in the pipline. Some the the main questions among many other important ones in this

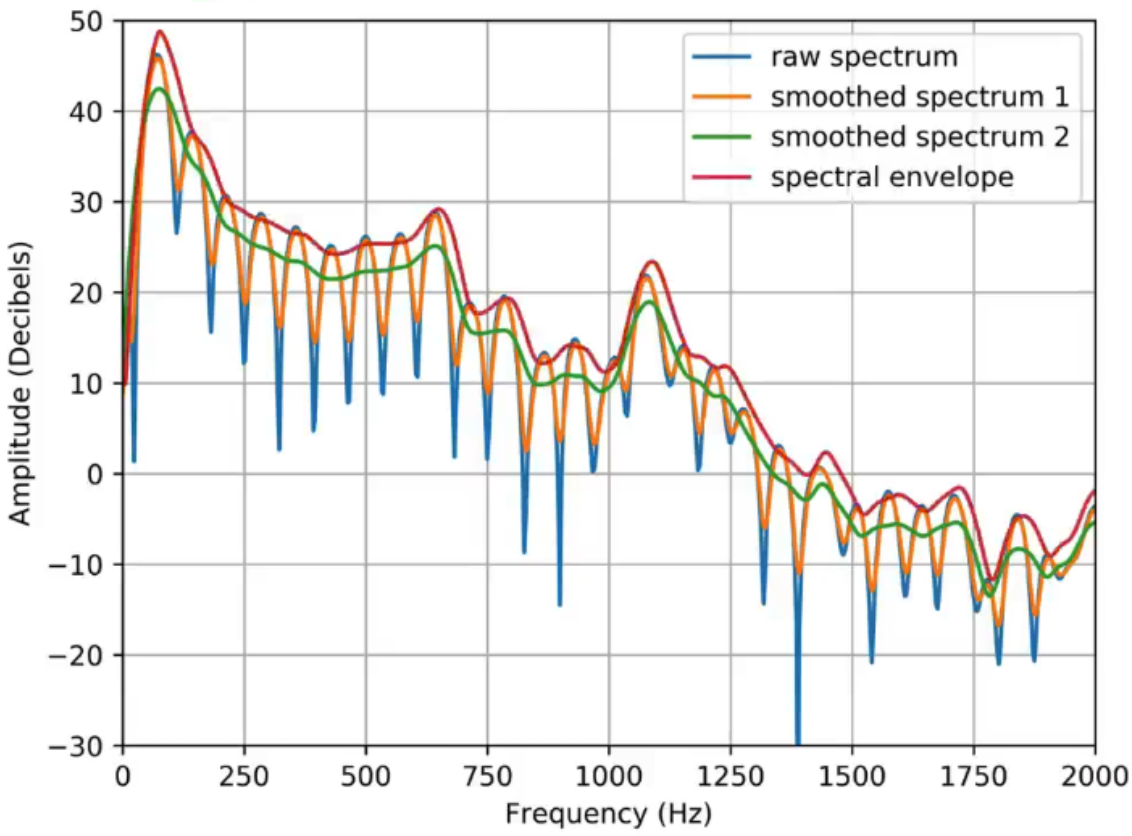


Figure 3.10: A complete process of spectral envelope estimation by cheaptrick algorithm is shown with the result of each process in different colors. The blue color spectrum is the result of Hanning window, The orange color depicts the result of first smoothing process, The green color shows the result the second smoothing process and finally, the red color spectrum shows the final spectrum. The whole process is carried out in cepstral domain but illustrated here in spectral domain to be intuitive.

context are:

- When to evaluate?
- Which aspect to evaluate?
- How to evaluate?

Concrete understanding of each component in the pipeline will help us, but expertise in all of the components is practically impossible, and that is the reason most papers rely on subjective measures in the evaluation phase only.This is difficult, even with expertise.

Some choices and decision must be made for evaluation purposes. Firstly, the purpose of each evaluation measure has to be chosen. For TTS systems, generally three types of tests exist: diagnostic tests to guide future development, comparative tests against another system

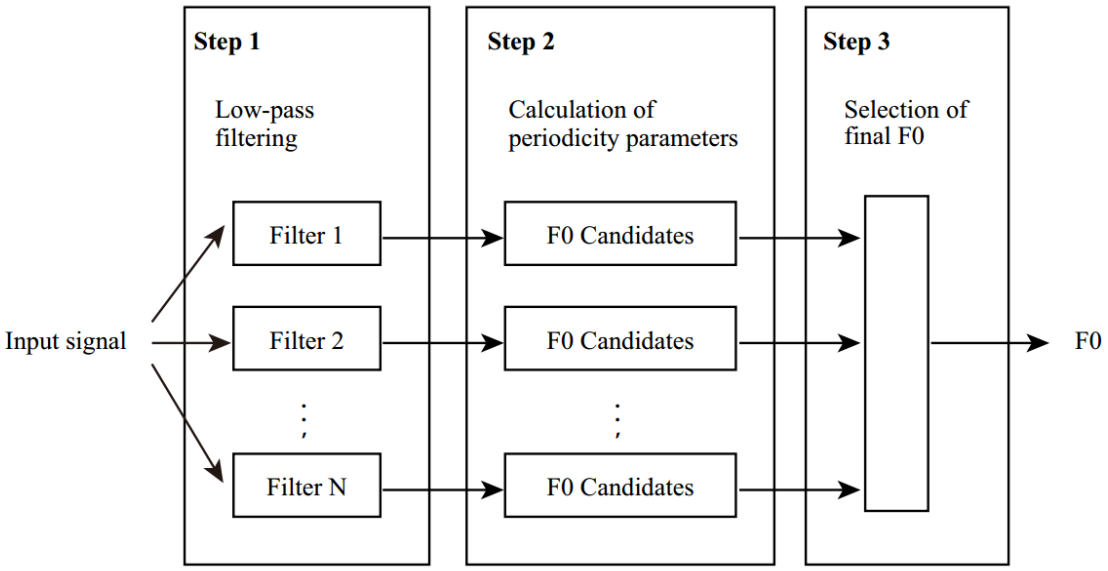


Figure 3.11: The process of fundamental frequency (F0) using DIO. Morise et al. (2009)

or a baseline, and pass/fail test for a product release. Secondly, we must consider the time for evaluation, either during development or after the completion of development. Thirdly, different aspects can be evaluated, such as intelligibility, naturalness, and speaker similarity. Finally, specifications of each evaluation has to be defined. For example: how should we design the tests, what materials have to be used, and what kind of objective and subjective measures should be considered.

In this work, for the evaluation of duration model, root mean sqared error (RMSE) on a millisecond scale is used. For the objective evaluation of the acoustic model, mel cepstral distortion (MCD) for Mel-frequency cepstral coefficients (MFCCs) in decibels [dB] , RMSE of the $F0$ prediction in Hertz [Hz], band aperiodicities (BAPs) distortion in [dB], and accuracy metric for voiced/unvoiced flag prediction in percentage [%] are used. For subjective evaluation, I perform MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor).

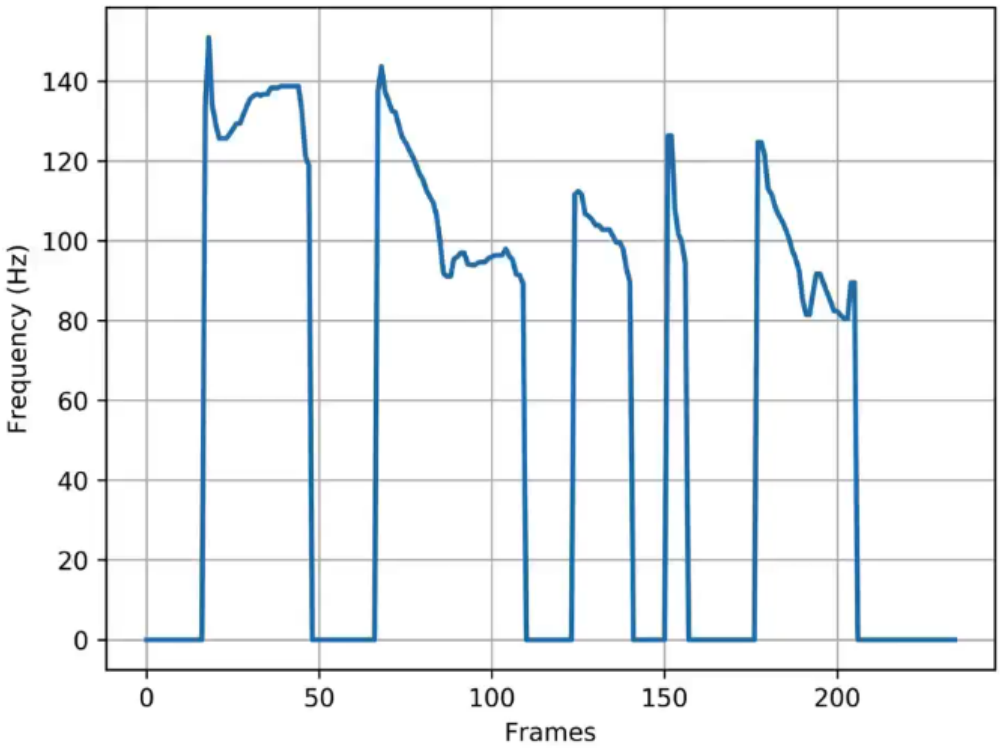


Figure 3.12: Fundamental frequency (F0) extracted by DIO algorithm is shown where unvoiced frames are zeros.

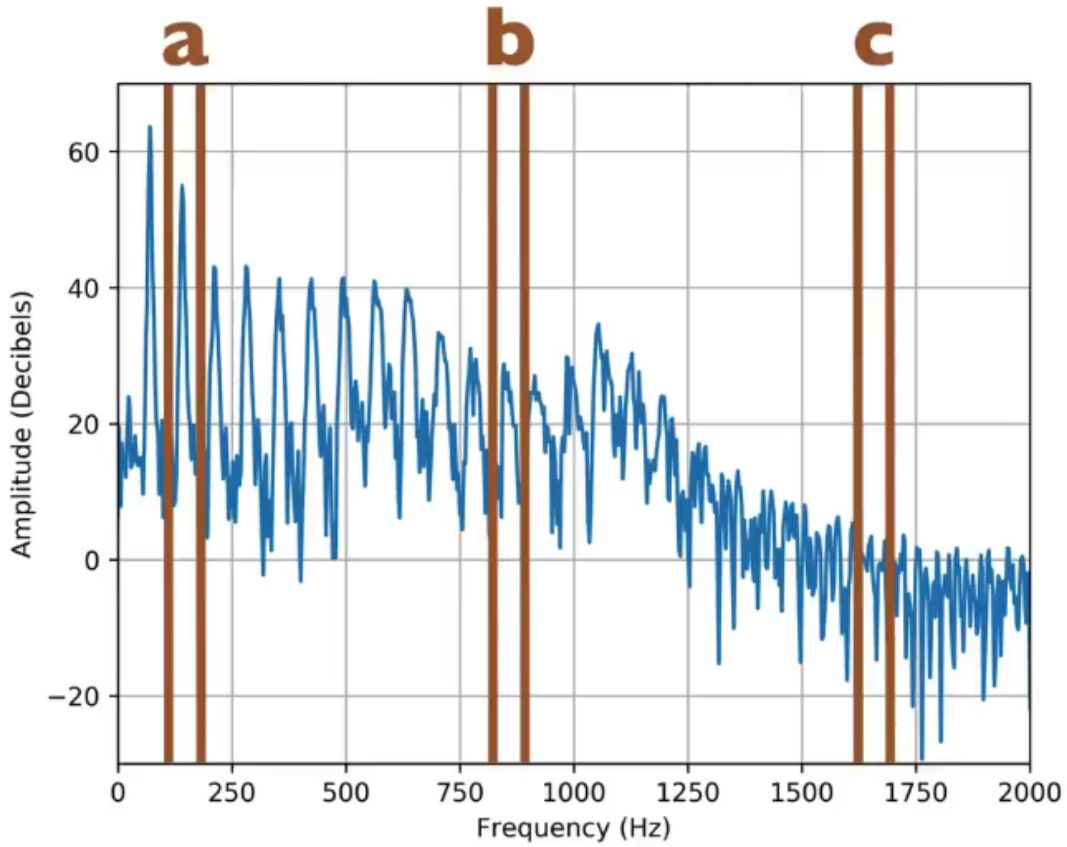


Figure 3.13: Three different bands of a, b, and c are shown. “a” for low in band means lower degree of randomness with more harmonics. “b” for more in band and “c” for highest in band, more random and less harmonic.

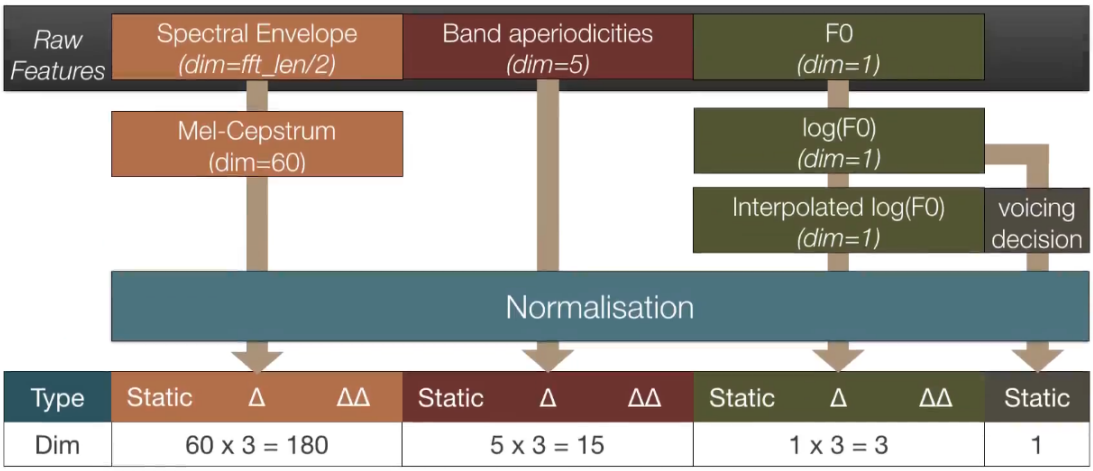


Figure 3.14: The figure illustrates the feature engineering process after the raw features are extracted by the vocoder. Then transformation from spectrum to cepstrum domain, From F0 to $logF0$ and interpolated $logF0$, addition of voicing decision information, normalization of all the features and the application of delta, and delta-delta operation along with the total number of parameters are shown.

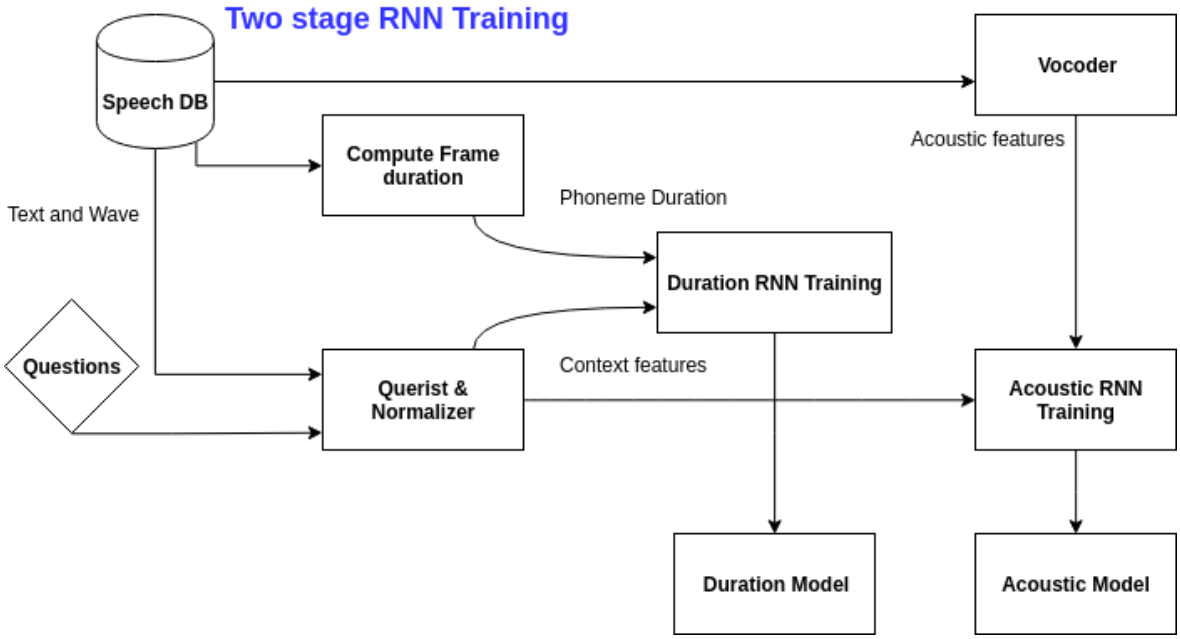


Figure 3.15: Training module architecture of RNN-LSTM text synthesis. Adapted from Wu et al. (2016).

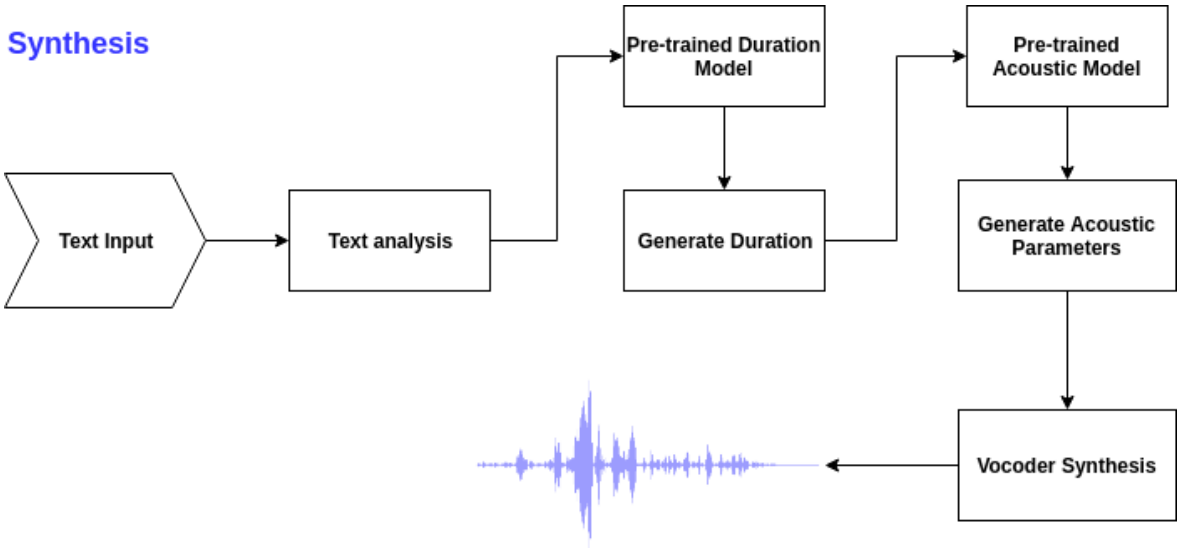


Figure 3.16: Synthesis module of two-stage RNN-LSTM model. Adapted from Wu et al. (2016).

Chapter 4

Experimental Results

This chapter is mainly focuses on the implementation details of Wolwala. It has three main parts: a detailed analysis of the process based on the methodology specified in Chapter 3 including a front-end analysis, a input features extraction and engineering, output features extraction and engineering, and RNN-LSTM model details followed by an objective and subjective evaluation of the developed system, and finally, a brief discussion.

4.1 Overview

To help the reader understand the flow of this chapter, I review some of the important points from Chapters 1, 2, and 3. For most prominent languages, researchers have focused on three basic components: a conventional front-end/text analysis component in which linguistic specifications are extracted, as shown in Figure 4.1, a machine learning component in which sequence to sequence regression is used, for example with HMMs or neural networks, and a digital signal processing/waveform synthesis module in which acoustic features are extracted for training and predicted features are processed to produce speech.

Normalization, tokenizing text, part of speech tagging, letter-to-sound mechanisms, phrase breaks, and intonations are among the important parts of the front-end component. Existing front-ends for languages that have them provide each of the above parts already. Developing each mentioned part for a new language itself is a large amount of work. For instance, to develop a reasonably good part-of-speech tagger, more than one million labelled tokens are required. This might take more than a year for two or more people.

To overcome this problem for the Pashto language and to avoid repetition, I usean unsupervised learning approach to substitute a conventional front-end. Some of the most important points of this approach are discussed later, while ignoring the parts that are same as these would be in a conventional pipeline as can easily be found in the literature.

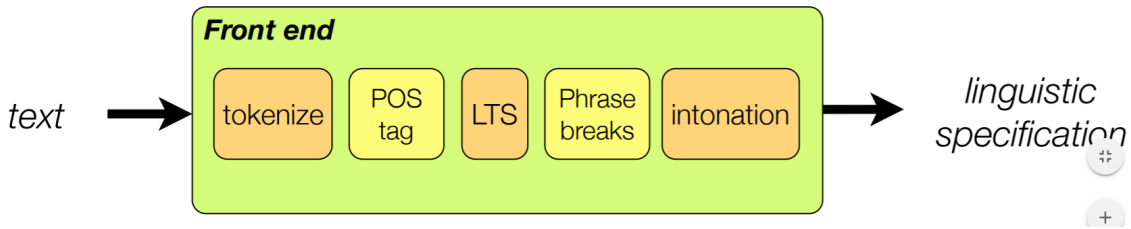


Figure 4.1: Conventional front-end for text-to-speech systems. Adopted from Taylor (2009).

4.2 Corpus creation

As many have stated, the new gold in the tech industry is cleaned and annotated data. Getting cleaned and annotated data is either difficult or very expensive. In many cases, for special purposes, its both difficult and expensive. Fortunately, there are many good people serving humanity. Open data communities and some technology companies are sharing datasets enabling everyone is allowed to make use of them. Unfortunately, after exhaustive search for large, medium, or small voice databases for the Pashto language as a whole and for text to speech or speech to text datasets particularly, I did not find any.

Keeping the issue of unavailability in mind, I developed my own dataset for speech synthesis and speech recognition purposes. I started by studying the standardized way of preparing a corpus for extraction of text and its recording. After almost two and half months of consecutive text extraction, normalization, and recording, a highly accurate dataset containing 5000 utterances and corresponding normalized text was produced. The total amount of recorded data is about 13.5 hours, which is a decent amount of data for text to speech purposes. Although the main motivation for creating the dataset is enrichment of Pastho language speech synthesis, it could also be used for automatic speech recognition and as a component in an end-to-end machine translation systems. Similarly, with an online dataset available free of cost for research purposes, native Pashtoon researchers may also turn their attention to these domains, resulting in nourishment of resources for the Pashto language.

It is worth mentioning that the dataset covers various contexts including sports, health, entertainment, and politics. election, war, and peace related issues were trending during the text extraction period. Based on my own observation, most of the frequently used words in day to day conversation of Pashto language can be found in the corpus, but words mostly used in a formal context do not appear frenquently enough. Thus, researchers using this dataset in the future would be requested to provide input to further enrich the dataset both in terms of quantity and quality.

4.3 Front-end analysis, input feature extraction, and engineering

Front-end analysis, input feature extraction, and engineering refer to analysis and labeling of raw text input and preparing it to be passed as input to a DNN model. Figure 4.1 depicts tokenization, part of speech tagging, letter to sound rules, phrase breaks, and intonations as the main parts of a front-end analysis component. But even before tokenization, normalization is the first step to be tackled. Normalization is not mentioned in most literature because it is an inherent part of the corpus creation process rather than a separate process. After front-end analysis, the result should be flattened in order to be passed to a neural network.

4.3.1 Normalization

As I prepared the corpus for Wolwala, I normalized the data manually during preparation. Most normalization problems in Pashto text were related to ordinal and cardinal numbers and dates. Similarly, a small amount of abbreviations and acronyms were found and transformed accordingly.

4.3.2 Tokenization

The general, simple and language-independent regular expression used to tokenize is

```
([\\p{L}||\\p{N}||\\p{M}]+)
```

This regular expression splits sentences based on Unicode characters into letters, numbers, and punctuation marks as shown in the `token_class` attribute of the token element in Figure 4.2. I find that this works fine for the Pashto language as well.

4.3.3 Distributional word vectors as part of speech (POS) tag substitutes

This section describes the use distributional word vectors as a substitute for a part-of-speech (POS) tagger for languages without POS taggers. The distributional word vector technique was previously implemented and tested by Watts (2013) for Romanian, Finnish, and English. The idea is to use simple statistics such as unigram and bigram frequencies. Unigram statistics provide information about high, mid, and low frequency words. Bigram statistics provide even richer information about the context of words, for example the knowledge that a specific word occurrs more/less frequently before another specific word. A matrix of left and right bigram co-occurrence counts with a handful of frequent words is created to represent the behaviour of a specific word in each context (200 to 300 words in practice). Obivously, this representation is large and noisy. To compress and de-noise it, singular-value decomposition (SVD), a factorization technique for real and complex matrices, is applied. The result is a low-dimensional word vector representation that captures regularities that are useful for text-to-speech synthesis.

4.3.4 Letter to sound

Unfortunately, no digital phonemic dictionary for Pashto is available yet, but as the number of letters (45) and phonemes (48) in the Pashto language are near to each other, a letter-based

```
<utt text="د افغانستان پارلماني ياكنو كې څه باندې دوه نيم زره نوماندان سيالي كوي."
waveform="ARS90.wav" utterance_name="ARS90" processors_used=",word_splitter" >
<token
    text="_END_"
    token_class="_END_"
    safetext="_END_" />
<token
    text="د"
    token_class="word"
    safetext="_ARABICLETTERDAL_" />
<token
    text=" "
    token_class="space"
    safetext="_SPACE_" />
<token
    text="افغانستان"
    token_class="word"
    safetext="_ARABICLETTERALEF__ARABICLETTERFEH__ARABICLETTERGHAIN
    __ARABICLETTERALEF__ARABICLETTERNOON__ARABICLETTERSEEN__
    ARABICLETTERTEH__ARABICLETTERALEF__ARABICLETTERNOON_" />
<token
    text=" "
    token_class="space"
    safetext="_SPACE_" />
<token
    text="پارلماني"
    token_class="word"
    safetext="_ARABICLETTERPEH__ARABICLETTERALEF__ARABICLETTERREH__ARABICLETTERLAM
    __ARABICLETTERMEEM__ARABICLETTERALEF__ARABICLETTERNOON__ARABICLETTERYEH_" />
.
.
.
    <token
    text="."
    token_class="punctuation"
    safetext="_FULLSTOP_" />
<token text="_END_" token_class="_END_" safetext="_END_"/>
</utt>
```

Figure 4.2: A snippet of tokenized text based on a general regular expression that for most left-to-right or right-to-left written languages in which the words are separated by spaces.

model in which letter names are used as names of speech modeling units can be used. In this approach, each letter is changed into its corresponding Unicode version and is used in the development of the system, as shown for example in Figure 4.3.

4.3.5 Forced alignment and silence detection

For forced alignment, the HTS force alignment toolkit is used to extract timing information such as silence in the utterance, the start of a letter, the end of a letter, and sub-phone information from the the utterance, and then the extracted information is appended into the front-end features file. A snippet is shown in Figure 4.4.

```
      has_silence="no"
      phrase_start="False"
      phrase_end="False">
  <segment pronunciation="_ARABICLETTERDAL_" start="0" end="410"> <state ... /> ...</segment>
</token>
<token text=" " token_class="space" ... />
<token text="افغانستان"
    token_class="word"
      safetext="_ARABICLETTERALEF__ARABICLETTERFEH__ARABICLETTERGHAIN__ARABICLETTERALEF__ARABICLETTERNOON
        __ARABICLETTERSEEN__ARABICLETTERTEH__ARABICLETTERALEF__ARABICLETTERNOON_"  ... >
  <segment pronunciation="_ARABICLETTERALEF_"     start="410" end="600"  ><state ... />... </segment>
  <segment pronunciation="_ARABICLETTERFEH_"      start="600" end="675"  ><state ... />... </segment>
  <segment pronunciation="_ARABICLETTERGHAIN_"    start="675" end="870"  ><state ... />... </segment>
  <segment pronunciation="_ARABICLETTERALEF_"     start="870" end="960"  ><state ... />... </segment>
  <segment pronunciation="_ARABICLETTERNOON_"     start="960" end="1005" ><state ... />... </segment>
  <segment pronunciation="_ARABICLETTERSEEN_"     start="1005" end="1135"><state ... />... </segment>
  <segment pronunciation="_ARABICLETTERTEH_"      start="1135" end="1170"><state ... />... </segment>
  <segment pronunciation="_ARABICLETTERALEF_"     start="1170" end="1260"><state ... />... </segment>
  <segment pronunciation="_ARABICLETTERNOON_"     start="1260" end="1345"><state ... />... </segment>
<token text=" " token_class="space" ... />
<token text="پارلماني"
    token_class="word"
    safetext="_ARABICLETTERPEH__ARABICLETTERALEF__ARABICLETTERREH__ARABICLETTERLAM__ARABICLETTERMEEM_
    _ARABICLETTERALEF__ARABICLETTERNOON__ARABICLETTERYEH_"
    vsm_d1="0.36166081978126635"
    vsm_d2="0.27573092263706295"
    vsm_d3="0.5041953501521392"
```

Figure 4.3: A snippet of letter-to-sound definitions from the file generated for the word "افغانستان" (Afghanistan), denoted by Unicode names along with start and end timing information from the utterance.

4.3.6 Phrase breaks

A naive but effective way for phrase break detection in languages where a separate component for phrase breaks is not available is using silences as a proxy for prosodic phrase breaks. The use of silence as a proxy for phrase break combining information from both natural language and speech eliminates the need for a separate phrase break identifier. Although it is not highly accurate, filtering with minimum silence duration can provide satisfactory results. A snippet of phrase break results based on the prosodic phrase breaks is shown in Figure 4.5

4.3.7 Input feature engineering

Input feature engineering refers to obtaining and flattening linguistic specifications, attaching contextual information to phones, encoding each context dependent phone as a vector, encoding as mostly binary feature, up-sampling using duration information, changing duration into the frame sequence, and adding fine-grained positional information to the sequence, and preparing it to be passed as input to a DNN model.

```
.
.
.
<token text="افغانستان" ... >
  <segment pronunciation="_ARABICLETTERALEF_" start="410" end="600">
    <state start="410" end="420"/>
    <state start="420" end="435"/>
    <state start="435" end="570"/>
    <state start="570" end="575"/>
    <state start="575" end="600"/>
  </segment>
  <segment pronunciation="_ARABICLETTERFEH_" start="600" end="675">
    <state start="600" end="615"/>
    <state start="615" end="650"/>
    <state start="650" end="655"/>
    <state start="655" end="660"/>
    <state start="660" end="675"/>
  </segment>
  <segment pronunciation="_ARABICLETTERGHAIN_" start="675" end="870">
    <state start="675" end="680"/>
    <state start="680" end="835"/>
    <state start="835" end="840"/>
    <state start="840" end="865"/>
    <state start="865" end="870"/>
  </segment>
  <segment pronunciation="_ARABICLETTERALEF_" start="870" end="960">
    <state start="870" end="875"/>
.
.
.
```

Figure 4.4: An example of HTS forced alignment, along with the starting information of a letter and five subphones for each letter.

4.4 Output (acoustic) feature extraction and engineering

In this research work, based on previous works, I use three features: the spectral envelope, the fundamental frequency, and band aperiodicities for the analysis of the output. I extract features using the WORLD vocoder. The spectral evelope is estimated using the CheapTrick algorithm, the fundamental frequency (F0) is estimated using the DIO algorithm, and band aperiodicities are estimated using D4C algorithm. All three of the mentioned algorithms are implemented by the WORLD vocoder, developed by Masanori in 2009 (Morise et al., 2016). Its standard, open-source, free, stable, and compatible with the Merlin library.

4.4.1 Spectral envelope estimation

A very high level overview of spectral envelope estimation using the CheapTrick algorithm (Morise, 2015) includes the following steps: designing a window function based on the idea of pitch synchronous analysis (Mathews et al., 1961) with a Hanning window of length of 3 pitch periods ($3T_0$) is used. The second step is smoothing the power spectrum by applying a moving average filter of size $(2/3)F0$ to get rid of spikes in the negative direction, which makes the signal more stable. After the first smoothing process, another smoothing process of $2F0$ is applied to that makes the curve look like a spectral envelope. Refer to Figure 4.6. The evelope needs to be lifted with respect to the moving average, Finally, a weighted sum

```
<phrase>
  <token text="خه" token_class="word" safetext="_ARABICLETTERHAHWITHTHREEDOTSABOVE__ARABICLETTERHEH_" ... >
    <segment pronunciation="_ARABICLETTERHAHWITHTHREEDOTSABOVE_" start="2890" end="3025">...</segment>
    <segment pronunciation="_ARABICLETTERHEH_" start="3025" end="3125">...</segment>
  </token>
  <token text=" " token_class="space" safetext="_SPACE_" silence_predicted="0" phrase_start="False"... />
  <token text="با ندی" token_class="word" safetext="_ARABICLETTERBEH__ARABICLETTERALEF__ARABICLETTERNOON..." ... >
    <segment pronunciation="_ARABICLETTERBEH_" start="3125" end="3205">... </segment>
  </token>
</phrase>
<token text=" " token_class="space" safetext="_SPACE_" start="3765" end="3995" has_silence="yes" ... >
  <segment pronunciation="sil" start="3765" end="3995">... </segment>
</token>
<phrase>
  <token text="دوه" token_class="word" safetext="_ARABICLETTERDAL__ARABICLETTERWAW__ARABICLETTERHEH_" ... >
  ...
  </token>
  <token text=" " token_class="space" safetext="_SPACE_" silence_predicted="0" phrase_start="False" .../>
  <token text="نیم" token_class="word" safetext="_ARABICLETTERNOON__ARABICLETTERFARSIYEH__ARABICLETTERMEEM_" ...>
  ...
  </token>
  <token text=" " token_class="space" safetext="_SPACE_" silence_predicted="0" phrase_start="False" ... />
  <token text="زره" token_class="word" safetext="_ARABICLETTERZAIN__ARABICLETTERREH__ARABICLETTERHEH_" ...>
  ...
  </token>
  <token text=" " token_class="space" safetext="_SPACE_" silence_predicted="0" phrase_start="False" ... />
  <token text="نوما ندان" token_class="word" safetext="_ARABICLETTERNOON__ARABICLETTERWAW__ARABICLETTERMEEM..." ..>
  ...
  </token>
</phrase>
```

Figure 4.5: Two phrases detected by silence as a proxy method combining information from natural language processing and speech signals are shown.

of a shifted version of the given curve is taken as the spectral envelope of the waveform.

4.4.2 Fundamental frequency estimation

To extract the fundamental frequency (F0) using DIO, an interval detection method that consists three steps is applied. The first step in the process is low-pass filtering with different cut-off frequencies. The second step is the calculation of the variances of negative and positive going zero-crossing intervals and the intervals between the successive peaks and dips (The Fundamentalness) and the final step the selection of lowest fundamental frequency (F0) based on the fundamentalness from all the candidates as the final fundamental frequency (F0). Figure 4.7 depicts the variation of fundamental frequency (F0) extracted using DIO algorithm.

4.4.3 Band aperiodities estimation

The final important acoustic feature is band aperiodicity or in simple words, the degreee of randomness in a certain band. It is calculated as the ratio between aperiodic and periodic energy, averaged over certain frequency bands. i.e. it is calculated in D4C by the division of “total power” over the “sine wave power”. In the Figure 3.13 three bands: lowest in band as “**a**”, more in band as “**b**”, and highest in band as “**c**” are shown. band aperiodicity shows

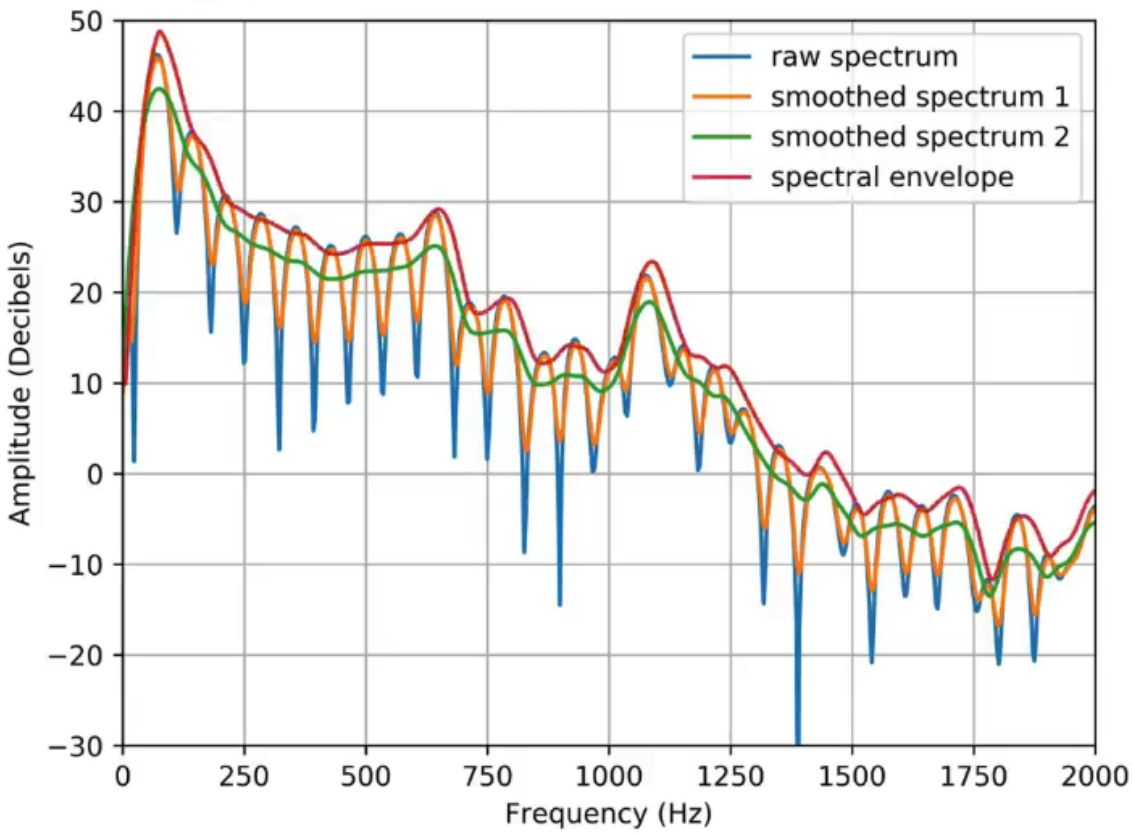


Figure 4.6: A complete process of spectral envelope estimation by cheaptrick algorithm is shown with the result of each process in different colors. The blue color spectrum is the result of Hanning window, The orange color depicts the result of first smoothing process, The green color shows the result the second smoothing process and finally, the red color spectrum shows the final spectrum. The whole process is carried out in cepstral domain but illustrated here in spectral domain to be intuitive.

that if the a signal is more random and has less harmonics the band aperiodicity value will be bigger and vice-versa.

To this point, the extraction process of three important acoustic feature is explained. The results of the process are raw vocoder features from the waveforms which are usually high in dimensions. i.e. the dimensions of extracted spectral envelope are $fft\_length/2$ which may be around 1000 to 2000 coefficients or more. an extra step to process the extracted information in such a way that is convenient in terms of number of coefficients and format to be passed to neural network is required. This process is called output feature engineering.

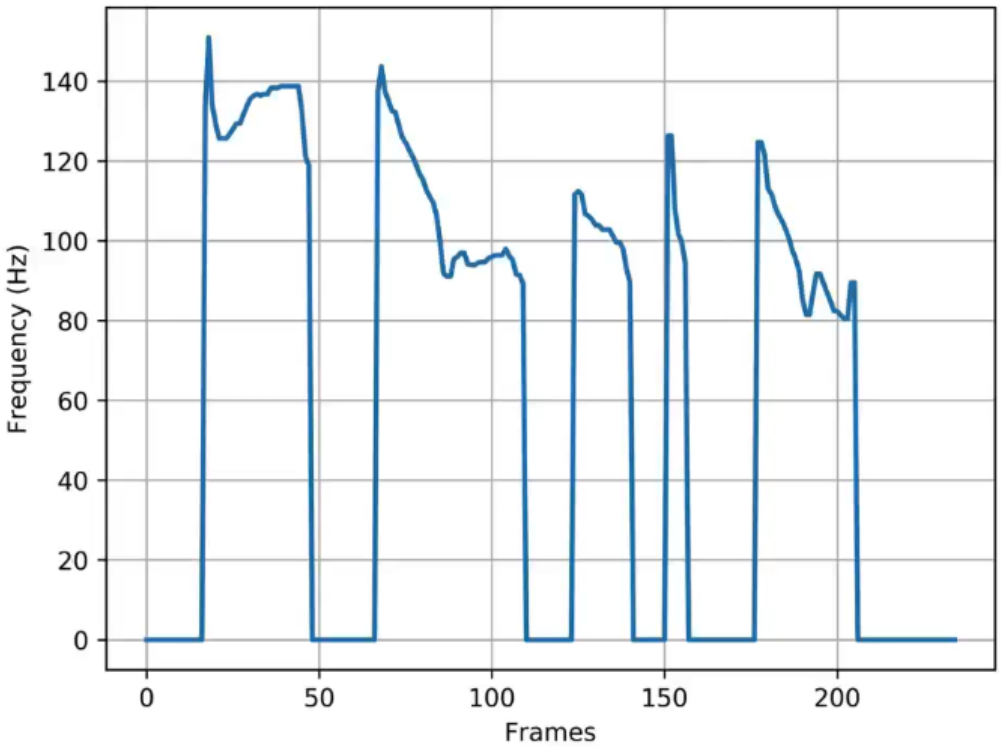


Figure 4.7: Fundamental frequency (F0) extracted by DIO algorithm is shown where unvoiced frames are zeros.

4.4.4 Output(acoustic) feature engineering

Output feature extraction and engineering refers to the extraction of a waveform specifications suitable for modeling that can be used to reconstruct the waveform. The main steps are extracting a spectral envelope, extracting a fundamental frequency ($F0$), extracting an aperiodic energy, computating mel-capstrum coeffiencts from the spectral envelope, computation of $_0$ from F_0, computation of interpolated $logF0$, making voicing decisions, and normalization. Finally, appending the delta, and delta-delta dynamic features. The result of output feature extraction are *.bap, *.lf0, and *.mgc for each utterance files.

4.5 Deep model details

To produce quality voice, two types of parameters are necessary:

- Prosodic parameters
- Acoustic parameters

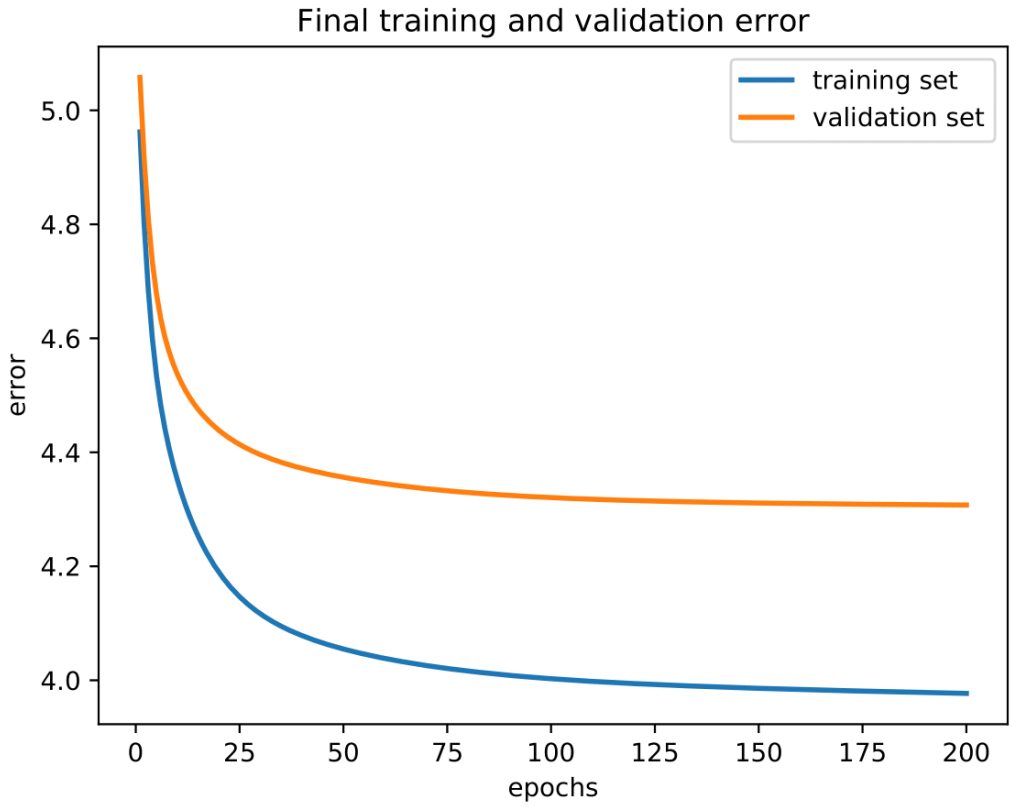


Figure 4.8: Convergence plot of Wolwala 0.5 duration model.

Prosodic parameters such as phoneme duration, stress, and pauses have high impact on the naturalness of of the voice. On the other hand, acoustic parameters such as spectral estimation used to generate the waveform by the vocoder, have high impact on both naturalness and intelligibility of the voice.

To acquire both types of parameters, we need two models: duration model and acoustic model. Duration model is required to predict the duration of the encoded linguistic feature input. An acoustic model is required to predict acoustic frame coefficients.

The recurrent neural network and its variants are well-suited to this task because of the following characteristics:

- We keep track of sequential context while reading input in a timely manner, one phoneme after another. Thus, each step encodes a label.
- The RNN's output layer improves the continuity of the predictions between frames in the acoustic prediction stage (Zen & Sak, 2015).

Although other architectures are also considered but for the first attempt it is a reasonable choice.

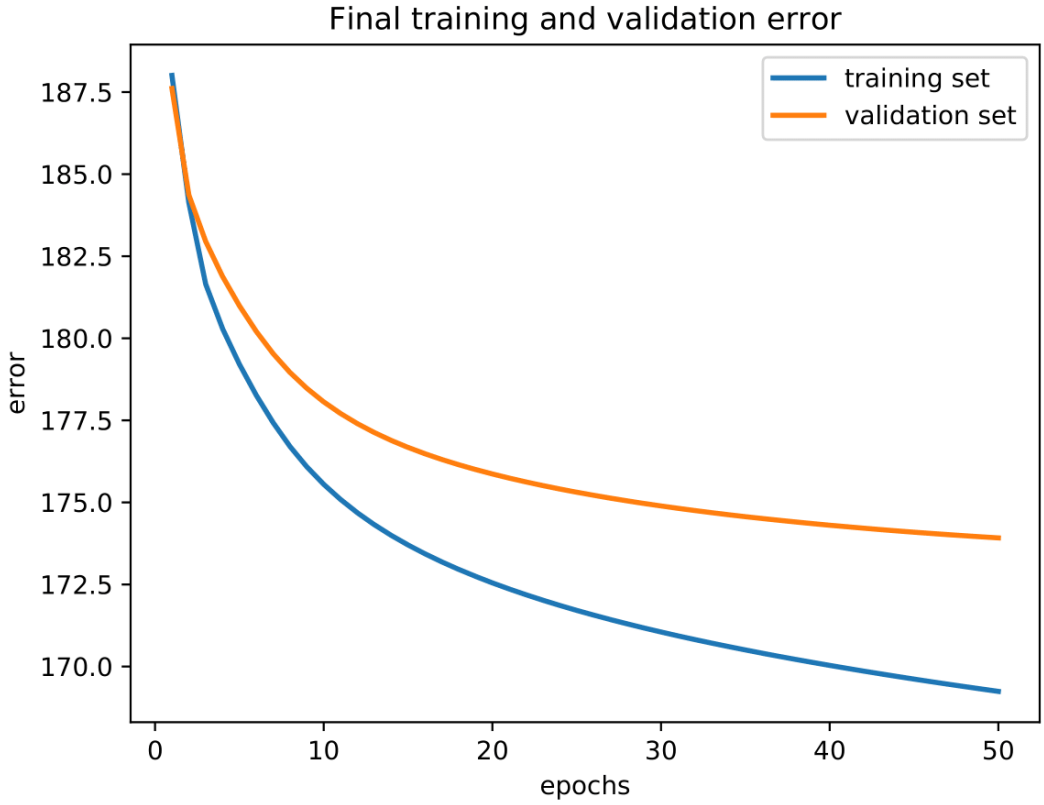


Figure 4.9: Convergence plot of Wolwala 0.5 acoustic model.

Several models have been developed and analyzed. The earliest model were developed to evaluate the possibility of generating synthesized speech. Different architectures were tried and hyper-parameters for every one of them were tuned. Some of the successful attempts are reported as follows: Wolwala 0.1 is among the first models trained with 300 utterances, the duration model has 3 hidden layers of size 512 of 'TANH' type, hidden activation function of 'tanh', learning rate of 0.002, batch size of 256, and 30 training epochs. The acoustic model has 6 hidden layers of size 1024 of 'TANH' type, hidden activation function of 'tanh', learning rate of 0.002, batch size of 256, and 100 training epochs.

Wolwala 0.3 is among the first models trained with 1000 utterances, the duration model has 3 hidden layers of size 512 of 'TANH' type, hidden activation function of 'tanh', learning rate of 0.002, batch size of 256, and 30 training epochs. The acoustic model has 6 hidden layers of size 1024 of 'TANH' type, hidden activation function of 'tanh', learning rate of 0.002, batch size of 256, and 100 training epochs.

Wolwala 0.5, the final model, is among the first models trained with 6000 utterances, the duration model has 3 hidden layers of size 512 of 'TANH' type, hidden activation function of 'tanh', learning rate of 0.002, batch size of 256, and 30 training epochs. Figure 4.8 shows convergence plot of Wolwala 0.5 duration model. The acoustic model has 6 hidden layers of

size 1024, first 5 of 'TANH' type and the last one as 'LSTM', hidden activation function of 'tanh', learning rate of 0.002, batch size of 256, and 100 training epochs. Figure 4.9 shows convergence plot of Wolwala 0.5 acoustic model. The rest of convergence graphs for training and validtion errors of each model are shown in the Appendix **??**.

4.6 Model evaluation

Despite the decades work in speech synthesis, the evaluation of speech synthesis systems is still a challenging task. Intelligility, naturalness and suitability for a specific domains remain the main criteria for speech synthesis, based on the judgement of the listeners on Lickert scale (Likert, 1932). For the evaluation of Wolwala, two categories of evaluation is conducted: objective evaluation and subjective evaluation.

4.6.1 Objective evaluation

For the evaluation of duration model, root mean sqared error (RMSE) in millisecond scale is used. RMSE for the duration model is defined as:

$$RMSE[ms] = \sqrt{\sum_{i=1}^{N-1}(Dur_t - \hat{D}ur_t)^2} \qquad \text{(Equation 4.1)}$$

Where N stands for the number of test phonemes used for evaluation. Table 4.1 shows the objective measures for the duration model.

For the objective evaluation of the acoustic model, mel ceptral distortion (MCD) for Mel-frequency cepstral coefficients (MFCCs) in decibels [dB] (Kubichek, 1993; Shannon et al., 2012), RMSE of the $F0$ prediction in Hertz [Hz], band aperiodicities (BAPs) distortion in [dB] (Cao et al., 2017), accuracy metric for voiced/unvoiced flag prediction in percentage [%] are used. Each of the measures are mathematically defined as follows:

$$MCD[dB] = (10\sqrt{2})/(T \ln 10) \sum_{t=0}^{T-1} \sqrt{\sum_{n=1}^{N}(C_{t,n} - \hat{C}_{t,n})^2} \qquad \text{(Equation 4.2)}$$

Where T is the number of test frames, N is the dimension of features, $\hat{C}_{t,n}$ are the predicted cepstral coefficients and $C_{t,n}$ are real cepstral coefficients.

$$RMSE[ms] = \sqrt{\sum_{t=0}^{T-1}(F0_t - \hat{F}0_t)^2} \qquad \text{(Equation 4.3)}$$

Where number of the test phonemes for evaluation are denoted as N, $\hat{F}0_t$ are predicted FOs, and $F0_t$ are real F0s.

$$BAP[dB] = (T/\log 10)\sum_{m=1}^{T}\sqrt{2\sum_{d=1}^{D}(C_{m,d} - \hat{C}_{m,d})^2} \qquad \text{(Equation 4.4)}$$

Where T is the number of test frames, $\hat{C}_{t,n}$ are the predicted voice and $C_{t,n}$ are original voice, m is the frame step (or time),D is the dimension of features, and d denotes dth dimension in frame m.

$$Acc[\%] = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \qquad \text{(Equation 4.5)}$$

Where TP stands for ture positives, TN for ture negatives, FP for false positives, and FN for false negatives. It is important to mention that the silences are removed and not included in the evaluation of both duration and acoustic model as their variation is huge and distorts the prediction. The Table 4.2 summarizes the results of objective evaluation of Wolwala.

	Model	RMSE [Hz]
Validation	RNN	4.39
Test	RNN	4.45

Table 4.1: Objective evaluation metrics forWolwala 0.5 in millisecond scale

Table 4.2: Objective evaluation metrics for Wolwala 0.4 (DNN model) and Wolwala 0.5 (RNN-LSTM model)

	Model	MCD [dB]	BAP [dB]	F0 RMSE [Hz]	V/UV [%]
Validation	DNN	4.698	0.365	10.145	11.40
Test	DNN	4.713	0.370	10.204	11.02
Validation	RNN-LSTM	4.695	0.320	9.850	11.03
Test	RNN-LSTM	4.583	0.342	9.986	11.01

4.6.2 Subject evaluation

As synthetic speech quality is an inherently a subjective experience and there is no guarantee that a model with little variation in objective evaluation will yield better synthesized speech. Among all the other models of tests, we conducted MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor), a well known test in speech community, listening test to subjectively evaluate the naturalness of the the synthesized speech. In MUSHRA test, 15 native Pashto listeners rated 20 sets of speech utterances from the evaluation set. For every sentence the listeners evaluated 3 different versions:

- Natural voice: original recording of the speaker.
- Vocoded voice: the acoustic parameters of the natural voice were were extracted with the same vocoder that is used for Wolwala system and reconstructed back. The main purpose of the vocoded voice is to evaluate the amount of loss of naturalness in the process of parametrization and reconstruction from the acoustic representation of the original signal.
- Wolwala voice: voice generated by the developed system.

Usually new systems are evaluated based on already existing systems for a specific language where the existing system will be considered as a baseline and newly developed system is compared to it. Evaluating the system against the natural speech especially in the first attempt will make the results look poorer and the improvements will not be as much exciting as it would have been in the first case. Figure 4.10 shows the MUSHRA scores for natural voice, vocoded voice and Wolwala voice. mean and variance table

4.7 Discussion

Technology as a tool to serve the world has been proven effective in many aspects particularly in cross-cultural communications. text-to-speech synthesis is an important ingredient in modern way of communication. The idea of machine translation for general purpose usage has helped many individual communicate to speaker of other languages with no knowledge of that particular language. For Pashto language, this was the first attempt to build a text-to-speech system from scratch. The main steps of the work includes: dataset creation, front-end analysis and input features extraction and engineering, output features extraction and engineering, model development, and evaluation. This approaches employed the recent one in the text-to-speech synthesis field using deep learning models. The results are significant. The intelligibility of the speech is high. However, the naturalness of the speech is comparatively lower, but it is as good as the synthesized speech of English generated by the similar techniques. Developing Pashto-specific components that are missing in the pipeline for the front-end and adding more quality data to the dataset will improve it even more. During the course of the model development some end-to-end models as discussed in Chapter 2 were also tried. The process is still in progress. However, due to the time limitation, end-to-end

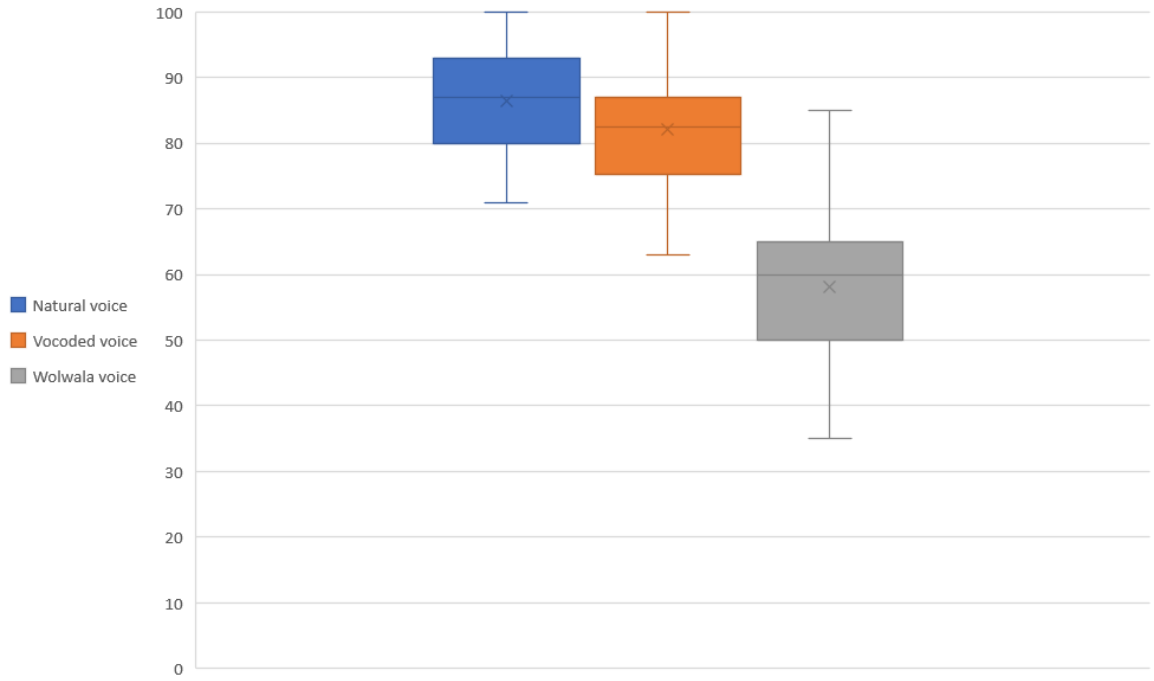


Figure 4.10: MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor) subjective evaluation score results for natural voice, vocoded voice, and Wolwala using WORLD vocoder.

deep models, as they take weeks to train, were ignored and not included in this work. I will report the results in my future work.

Chapter 5

Conclusion and Recommendations

This chapter touches concisely the text to speech synthesis along with its latest trends , an overview of the approach taken to develop Wolwala, the reasoning behind choosing that specific approach, and a few points regarding the evaluation of the developed system are presented, a brief introduction of the developed dataset for Pashto language that could be used for text to speech and speech to text is given, and lastly, some recommendations and future research directions are pointed out.

5.1 Conclusion

The idea of human like artificial production of intelligible and natural sounding speech has a deep root in the history that dates back to thousands of years but successful recorded attempts that could emulate some segments of such systems dates back to the late 18^{th} century in forms of mechanical systems. In late 19^{th} century researcher were able to produce complete systems. However the quality of speech was poor and robotic. In the first decade of 21^{th} century, two techniques namely parameteric speech synthesis and unit-selection of concatenative speech synthesis among all the other techniques dominated the text to speech synthesis field mainly because the researchers were able to yield better results.

Parametric speech synthesis's main advantage over the others is its flexibility that comes in many forms such as changing the specification of voice, its speaking style, and emotions. Adaptation, interpolation, eigenvoice, and multiple regression are some of the techniques that deliver this flexibility (Zen et al., 2009) but it suffers from lower speech quality that is inherited in three main factors: acoustic model accuracy, vocoder and over-smoothing.

On the other hand, unit-selection from concatenative speech synthesis family is producing the most natural speech in comparison to the parametric speech synthesis that turned the attention of researcher to itself. The result of this attention was the development of many text to speech systems mostly for specific domains with the underlying unit-selection technique but fine tuning the system is manual and hard. Similarly, adapting the unit-selection based text to speech systems to general context is challenging as preparing huge databases for each possible context in the smallest unit level is impractical . The unit-selection and parametric speech synthesis remained competing each other, as one could produce better intelligible speech and the other could produce more natural speech but both have deficiencies which lead to the idea of hybrid speech synthesis which is the use of both techniques as complementary to each others deficiencies.

Later, with the rise of computational power and deep learning techniques many fields of artificial intelligence revived and started solving the problems with a different approach. Researchers in the speech domain shares the same story as of others and tried similar tech-

niques especially parametric speech synthesis approach with different algorithms, enormous amount of data, and tremendous amount of computational resources. The results were outstanding. Researchers reported significant amount of improvements over the previous works in text to speech synthesis. Although, near natural speech was achieved with hundreds of hours of speech and its corresponding text, but practically it is still bound to some limited domains that the dataset represents.

The first attempts of deep models focused on replacing the traditional machine learning algorithms such as HMMs and decision trees that were prominently used for text to speech synthesis with many different deep learning architectures from fully connected deep neural networks (DNN) to recurrent neural networks (RNN) and its variations such as Long short term memory (LSTM) and Bi-directional LSTMs, deep belief networks (DBN), deep mixture density networks (MDN) and others. Most of them reported significant improvements over traditional parametric speech synthesis.

The later trend of deep learning turned the attention of researchers to replace the traditional pipeline into a complete end-to-end model. Many successful attempts such as CHAR2WAV, Wavenet, fastWavenet, parallelWavenet, tacotron 1 and 2, deep voice 1, 2, and 3 reported mean opinion score (MOS) of above than 3.5 out of 5 for English language in text to speech synthesis.

The main objective of this research was to build a text to speech synthesis for Pastho language. Pashto language belongs to a totally different category of languages in comparison to English. Its phonetics and phonology, morphology, syntax, semantics and pragmatics are completely different. In text to speech synthesis, the concentration is more on the writing system and phonology of a language compared to the other aspects. It takes a long time even for humans to master these two aspects. Not only peculiarities of a language even general rules of a specific language could rarely be applied to a language that does not share the same roots at least in a near past. Thus the previous work for English language could not be applied directly into Pastho text to speech synthesis. This problem could be approached in one of two ways: either to develop novel and Pashto-specific techniques or to make use of some general language independent concepts that could solve many cases, if not all. As the feature engineering part of the front end component in the traditional approach is highly language dependent, the second approach was chosen and applied. Obviously, tools developed specially for a specific language would bring improvements but each of the components itself requires a separate research work and could not be tackled in the limited amount time besides this work itself. Hence, In future research direction, I will point out and recommend researchers to develop some of the important components for Pashto language.

The second concern was the unavailability of the dataset (text and its corresponding utterance for text to speech and speech to text purposes) for Pashto language. Developing the dataset is a simple task in its nature but it is repetitive and exhaustive to extract, normalize, and record thousands of utterances. With the importance of the dataset in mind and after the discussion with researchers, I started developing the dataset, adhering to the standard procedures, preparing some professional tools with a calm and quiet environment for recording. I extracted and normalized the script mainly from news broadcasting agencies such as the British broadcasting corporation (BBC) and the voice of America (VOA). The total amount

reaches to 5000 (five thousand) recorded utterances of varied lengths from 3 to 45 seconds that sums up to 13.5 hours of transcribed speech which is decent amount of data for text to speech synthesis. The quality of the recorded speech from the environmental perspective was perfect, but the quality of my voice, as I am not a professional speaker, was comparatively a little lower. As the parliament elections were the hot topics on the news during the extraction process, most of the text covers the context of elections besides war and peace issues along with some sports and health related context. The recording tends to sound like the voice of a news anchor. The synthesized speech is also sounds similar to the news anchor. The main motivation behind developing the dataset was to encourage and redirect the attention of researchers specifically to the speech related fields of artificial intelligence such as speech synthesis and speech recognition for Pashto language. As the unavailability of the dataset was one of the factors that researcher were deviating from research in this domain for Pashto language.

After the development of Wolwala Pashto text to speech synthesis system, it was evaluated with objective and subjective measures. The results, as it was the first attempt, were almost the same as the systems developed for English with the same techniques. The same process with more and high quality data could achieve better intelligible and natural synthesized speech.

5.2 Recommendations and future research directions

Pashto language has not received enough amount of attention due to the limited number of native pashtoon researchers mainly because of wars and crisis in the region for last 50 years. There exists a huge gap for research on every aspect of technology as a whole and speech related domains of artificial intelligence as comparatively small areas to dive in. Unfortunately, no significant modules, specifically for Pashto, exist even the smaller ones. Some suggestions for the Pashto language depended modules should be developed are as follows:

- Development of text preprocessor to tokenize and normalize Pastho text, pronunciation dictionary, part of speech tagger, letter to sound rules, phrase and sentence identifiers are a few among many other that could be tackled by independent researchers.

Similarly, It is not necessary to imitate exactly the same approach that has been followed for English language. Researcher could come up with novel ideas in the following specialized areas:

- Attempts to discover alternative linguistic representations may yield better results that could not only be used for a specific language but might be able to map the linguistic representation of many similar languages.
- Another direction is to look for ways to extract the semantic features directly from the text and predict the acoustic features out of it. This research direction is aligned with work of Jauk et al. (2016)

- Another research direction is exploring the concept of combining unit selection, as it has the most natural synthesized speech for its best candidates but is suffering from joins, with neural networks techniques together as Hybrid text to speech synthesis.
- The end-to-end deep learning approach of text to speech systems tends to replicate the conventional pipeline of text to speech synthesis, except from some works that are using spectrograms directly rather than speech signals, and gets more and more complicated. A vast area of coming up with a novel and simple architectures is open that might be able to produce quality speech.
- Transfer learning is another possibility to be explored. Transfer learning could be applied to similar languages or at least similar in writing and phonology, such as Arabic, Persian, Urdu, Pashto, etc, for text to speech synthesis.

Apart from the text to speech domain, some other systems to complement the whole process of communication between two or more languages through deep learning are:

- Developing speech to text/speech recognitions systems for Pashto language could be the next step that can be developed with the current dataset.
- Developing automatic machine translation for Pashto language that would completed an end-to-end process of voice translation from one language to another is the next module in the middle of text to speech synthesis and speech to text systems but a separate multilingual or at least a bilingual, corpus is required to be developed for it.

As a native Pashto speaker and researcher, I will remain committed to develop and enhance these modules for Pashto langauage and serve humanity by focusing on novelties in my specialized domains.

References

Achanta, S., Godambe, T., & Gangashetty, S. V.(2015). An investigation of recurrent neural network architectures for statistical parametric speech synthesis. In *Sixteenth annual conference of the international speech communication association.*

Agiomyrgiannakis, Y.(2015). Vocaine the vocoder and applications in speech synthesis. In *Acoustics, speech and signal processing (icassp), 2015 ieee international conference on* (pp. 4230–4234).

Bahdanau, D., Cho, K., & Bengio, Y.(2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bengio, Y., Boulanger-Lewandowski, N., & Pascanu, R. (2013). Advances in optimizing recurrent networks. In *2013 ieee international conference on acoustics, speech and signal processing* (pp. 8624–8628).

Beutnagel, M., Conkie, A., Schroeter, J., Stylianou, Y., & Syrdal, A. (1999). The at&t next-gen tts system. In *Joint meeting of acoustical society of america , european acoustics association, and german annual conference on acoustics* (Vol. 1, pp. 18–24).

Black, A. W., & Taylor, P. A. (1997). Automatically clustering similar units for unit selection in speech synthesis.

Bonafonte, A., Nogueiras, A., & Rodriguez-Garrido, A. (1996). Explicit segmentation of speech using gaussian models. In *Spoken language, 1996. icslp 96. proceedings., fourth international conference on* (Vol. 2, pp. 1269–1272).

Cao, B., Kim, M. J., Santen, J. P. van, Mau, T., & Wang, J.(2017). Integrating articulatory information in deep learning-based text-to-speech synthesis. In *Interspeech* (pp. 254–258).

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L.(2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, S.-H., Hwang, S.-H., & Wang, Y.-R. (1998). An rnn-based prosodic information synthesizer for mandarin text-to-speech. *IEEE transactions on speech and audio processing*, *6*(3), 226–239.

Coto-Jiménez, M., & Goddard-Close, J.(2016). Lstm deep neural networks postfiltering for improving the quality of synthetic voices. In *Mexican conference on pattern recognition* (pp. 280–289).

Darmesteter, J.(1888). 1890. *Chants populaires des Afghans*.

Dempster, A. P., Laird, N. M., & Rubin, D. B.(1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1–38.

Donovan, R. E., & Woodland, P. C. (1995). Improvements in an hmm-based speech synthesiser. In *Fourth european conference on speech communication and technology.*

Dutilleux, P.(1990). An implementation of the algorithme à trous to compute the wavelet transform. In *Wavelets* (pp. 298–304). Springer.

Dutoit, T.(1997). *An introduction to text-to-speech synthesis* (Vol. 3). Springer Science & Business Media.

Dutoit, T., & Stylianou, Y.(2003). Text-to-speech synthesis. *Handbook of computational linguistics*, 323–338.

Fan, Y., Qian, Y., Soong, F. K., & He, L.(2015). Multi-speaker modeling and speaker adaptation for dnn-based tts synthesis. In *Acoustics, speech and signal processing (icassp), 2015 ieee international conference on* (pp. 4475–4479).

Fernandez, R., Rendel, A., Ramabhadran, B., & Hoory, R. (2014). Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks. In *Interspeech* (pp. 2268–2272).

Flanagan, J. L. (1972). Speech synthesis. In *Speech analysis synthesis and perception* (pp. 204–276). Springer.

Flanagan, J. L. (1973). *Speech synthesis* (Vol. 3). Dowden Hutchinson and Ross.

Freij, G., & Fallside, F. (1988). Lexical stress recognition using hidden markov models. In *Acoustics, speech, and signal processing, 1988. icassp-88., 1988 international conference on* (pp. 135–138).

Fukada, T., Tokuda, K., Kobayashi, T., & Imai, S. (1992). An adaptive algorithm for mel-cepstral analysis of speech. In *1992 ieee international conference on acoustics, speech, and signal processing, 1992. international conference on. acoustics, speech, and signal processing (icassp)-92.,* (Vol. 1, pp. 137–140).

Gibiansky, A., Arik, S., Diamos, G., Miller, J., Peng, K., Ping, W., et al. (2017). Deep voice 2: Multi-speaker neural text-to-speech. In *Advances in neural information processing systems* (pp. 2962–2970).

Hawley, M. S., Cunningham, S. P., Green, P. D., Enderby, P., Palmer, R., Sehgal, S., et al. (2013). A voice-input voice-output communication aid for people with severe speech impairment. *IEEE Transactions on neural systems and rehabilitation engineering*, *21*(1), 23–31.

Henderson, M. M. (1983). Four varieties of pashto. *Journal of the American Oriental Society*, 595–597.

Hirai, T., & Tenpaku, S. (2004). Using 5 ms segments in concatenative speech synthesis. In *Fifth isca workshop on speech synthesis.*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Holschneider, M., Kronland-Martinet, R., Morlet, J., & Tchamitchian, P. (1990). A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets* (pp. 286–297). Springer.

Hu, Q., Stylianou, Y., Maia, R., Richmond, K., Yamagishi, J., & Latorre, J. (2014). An investigation of the application of dynamic sinusoidal models to statistical parametric speech synthesis. In *Fifteenth annual conference of the international speech communication association.*

Hu, Q., Wu, Z., Richmond, K., Yamagishi, J., Stylianou, Y., & Maia, R. (2015). Fusion of multiple parameterisations for dnn-based sinusoidal speech synthesis with multi-task learning. In *Sixteenth annual conference of the international speech communication association.*

Huang, X., Acero, A., Adcock, J., Hon, H.-W., Goldsmith, J., Liu, J., et al. (1996). Whistler: A trainable text-to-speech system. In *Spoken language, 1996. international conference on spoken language processing (icslp) 96. proceedings., fourth international conference on* (Vol. 4, pp. 2387–2390).

Huang, X., Acero, A., Hon, H.-W., & Reddy, R. (2001). *Spoken language processing: A guide to theory, algorithm, and system development* (Vol. 1). Prentice hall PTR Upper Saddle River.

Hunt, A. J., & Black, A. W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *Acoustics, speech, and signal processing, 1996. icassp-96. conference proceedings., 1996 ieee international conference on* (Vol. 1, pp. 373–376).

Imai, S. (1983). Cepstral analysis synthesis on the mel frequency scale. In *Acoustics, speech, and signal processing, ieee international conference on icassp'83.* (Vol. 8, pp. 93–96).

Ipsic, I., & Martincic-Ipsic, S. (2006). Croatian hmm-based speech synthesis. *Journal of Computing and Information Technology*, *14*(4), 307–313.

ISHIMATSU, Y. (2001). Investigation of state duration model based on gamma distribution for hmm-based speech synthesis. *IEICE Technical Report*, SP2001–81.

Jauk, I., Bonafonte, A., & Pascual, S. (2016). Acoustic feature prediction from semantic features for expressive speech using deep neural networks. In *2016 24th european signal processing conference (eusipco)* (pp. 2320–2324).

Jensen, U., Moore, R. K., Dalsgaard, P., & Lindberg, B. (1994). Modelling intonation contours at the phrase level using continuous density hidden markov models. *Computer Speech & Language*, *8*(3), 247–260.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016). Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410.*

Kang, S., Qian, X., & Meng, H. (2013). Multi-distribution deep belief network for speech synthesis. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on* (pp. 8012–8016).

Kawai, H., & Tsuzaki, M. (2002). Study on time-dependent voice quality variation in a large-scale single speaker speech corpus used for speech synthesis. In *Speech synthesis, 2002. proceedings of 2002 ieee workshop on* (pp. 15–18).

King, S. (2011). An introduction to statistical parametric speech synthesis. *Sadhana*, *36*(5), 837–852.

Klatt, D. H. (1987). Review of text-to-speech conversion for english. *The Journal of the Acoustical Society of America*, *82*(3), 737–793.

Kubichek, R. (1993). Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of ieee pacific rim conference on communications computers and signal processing* (Vol. 1, pp. 125–128).

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Ling, Z.-H., & Wang, R.-H. (2006). Hmm-based unit selection using frame sized speech segments. In *Ninth international conference on spoken language processing.*

Lu, H., King, S., & Watts, O. (2013). Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis. In *Eighth isca workshop on speech synthesis.*

Mathews, M., Miller, J. E., & David Jr, E. (1961). Pitch synchronous analysis of voiced sounds. *The Journal of the Acoustical Society of America*, *33*(2), 179–186.

Mizutani, N. (2002). Concatenative speech synthesis based on hmm. In *Proc. autumn meeting of asj, 2002.*

Morise, M. (2015). Cheaptrick, a spectral envelope estimator for high-quality speech synthesis. *Speech Communication*, *67*, 1–7.

Morise, M., Kawahara, H., & Katayose, H. (2009). Fast and reliable f0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech. In *Audio engineering society conference: 35th international conference: Audio for games.*

Morise, M., Yokomori, F., & Ozawa, K.(2016). World: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, *99*(7), 1877–1884.

Mporas, I., Lazaridis, A., Ganchev, T., & Fakotakis, N. (2009). Using hybrid hmm-based speech segmentation to improve synthetic speech quality. In *Informatics, 2009. pci'09. 13th panhellenic conference on* (pp. 118–122).

Oord, A. van den, Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. (2016). Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems* (pp. 4790–4798).

Oord, A. van den, Kalchbrenner, N., & Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*.

Oord, A. v. d., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., et al. (2017). Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*.

Paine, T. L., Khorrami, P., Chang, S., Zhang, Y., Ramachandran, P., Hasegawa-Johnson, M. A., et al. (2016). Fast wavenet generation algorithm. *arXiv preprint arXiv:1611.09482*.

Ping, W., Peng, K., & Chen, J.(2018). Clarinet: Parallel wave generation in end-to-end text-to-speech. *arXiv preprint arXiv:1807.07281*.

Ping, W., Peng, K., Gibiansky, A., Arik, S. O., Kannan, A., Narang, S., et al. (2017). Deep voice 3: Scaling text-to-speech with convolutional sequence learning. *arXiv preprint arXiv:1710.07654*.

Qian, Y., Fan, Y., Hu, W., & Soong, F. K. (2014). On the training aspects of deep neural network (dnn) for parametric tts synthesis. In *Acoustics, speech and signal processing (icassp), 2014 ieee international conference on* (pp. 3829–3833).

Ross, K., & Ostendorf, M.(1994). A dynamical system model for generating f0 for synthesis. In *The second esca/ieee workshop on speech synthesis.*

Sagisaka, Y., Kaiki, N., Iwahashi, N., & Mimura, K. (1992). Atr μ-talk speech synthesis system. In *Second international conference on spoken language processing.*

Sakai, S., & Shu, H. (2005). A probabilistic approach to unit selection for corpus-based speech synthesis. In *Ninth european conference on speech communication and technology.*

Santen, J. P. v., & Sproat, R. W. (1999). High-accuracy automatic segmentation. In *Sixth european conference on speech communication and technology.*

Schroeder, M. R. (1993). A brief history of synthetic speech. *Speech Communication*, *13*(1-2), 231–237.

Segi, H., Takagi, T., & Ito, T. (2004). A concatenative speech synthesis method using context dependent phoneme sequences with variable length as search units. In *Fifth international speech communication association (isca) workshop on speech synthesis.*

Shannon, M., Zen, H., & Byrne, W. (2012). Autoregressive models for statistical parametric speech synthesis. *IEEE Transactions on Audio, Speech, and Language Processing*, *21*(3), 587–597.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., et al. (2018). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 4779–4783).

Shi, Y., Chang, E., Peng, H., & Chu, M. (2002). Power spectral density based channel equalization of large speech database for concatenative tts system. In *Seventh international conference on spoken language processing.*

Strabo, Jones, .-., Horace Leonard, & Sterrett, .-., J. R. Sitlington (John Robert Sitlington). (1917). *The geography of strabo* [Book; Book/Illustrated]. London (England) : W. Heinemann ; New York (New York) : Putnam's sons. (Greek text, with English translation on opposite pages)

Stylianou, Y.(1999). Assessment and correction of voice quality variabilities in large speech databases for concatenative speech synthesis. In *Acoustics, speech, and signal processing, 1999. proceedings., 1999 ieee international conference on* (Vol. 1, pp. 377–380).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Swietojanski, P., & Renals, S. (2014). Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. In *Spoken language technology workshop (slt), 2014 ieee* (pp. 171–176).

Tachiwa, W., & Furui, S. (1999). A study of speech synthesis using hmms. In *Proc. spring meeting of asj* (pp. 239–240).

Taylor, P.(2009). *Text-to-speech synthesis*. Cambridge university press.

Tokuda, K., Masuko, T., Miyazaki, N., & Kobayashi, T. (2002). Multi-space probability distribution hmm. *IEICE TRANSACTIONS on Information and Systems*, *85*(3), 455–464.

Tokuda, K., Nankaku, Y., Toda, T., Zen, H., Yamagishi, J., & Oura, K.(2013). Speech synthesis based on hidden markov models. *Proceedings of the IEEE*, *101*(5), 1234–1252.

Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., & Kitamura, T.(2000). Speech parameter generation algorithms for hmm-based speech synthesis. In *Acoustics, speech, and signal processing, 2000. icassp'00. proceedings. 2000 ieee international conference on* (Vol. 3, pp. 1315–1318).

Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., et al. (2016). Wavenet: A generative model for raw audio. In *Ssw* (p. 125).

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., et al. (2017). Tacotron: A fully end-to-end text-to-speech synthesis model. *arXiv preprint*.

Watts, O. S.(2013). Unsupervised learning for text-to-speech synthesis.

Wu, Z., & King, S. (2015). Minimum trajectory error training for deep neural networks, combined with stacked bottleneck features. In *Sixteenth annual conference of the international speech communication association.*

Wu, Z., & King, S.(2016). Investigating gated recurrent neural networks for speech synthesis. *arXiv preprint arXiv:1601.02539*.

Wu, Z., Swietojanski, P., Veaux, C., Renals, S., & King, S.(2015). A study of speaker adaptation for dnn-based speech synthesis. In *Sixteenth annual conference of the international speech communication association.*

Wu, Z., Valentini-Botinhao, C., Watts, O., & King, S.(2015). Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *Acoustics, speech and signal processing (icassp), 2015 ieee international conference on* (pp. 4460–4464).

Wu, Z., Watts, O., & King, S. (2016). Merlin: An open source neural network speech synthesis system. *Proc. SSW, Sunnyvale, USA*.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., & Kitamura, T.(1998). Duration modeling for hmm-based speech synthesis. In *Fifth international conference on spoken language processing.*

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., & Kitamura, T. (1999). Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. In *Sixth european conference on speech communication and technology.*

Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Ze, H., Senior, A., & Schuster, M. (2013). Statistical parametric speech synthesis using deep neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on* (pp. 7962–7966).

Zen, H. (2006). An example of context-dependent label format for hmm-based speech synthesis in english. *The HTS CMUARCTIC demo*, *133*.

Zen, H., & Sak, H. (2015). Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *Acoustics, speech and signal processing (icassp), 2015 ieee international conference on* (pp. 4470–4474).

Zen, H., & Senior, A. (2014). Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In *Acoustics, speech and signal processing (icassp), 2014 ieee international conference on* (pp. 3844–3848).

Zen, H., Tokuda, K., & Black, A. W. (2009). Statistical parametric speech synthesis. *Speech Communication*, *51*(11), 1039–1064.

Appendix A

.. TITLE HERE ..

Section Name

Table A.2:: Parameters of RNN architecture based duration model.

Begining of Table A.2	
Parameters	Values
mgc	
in_dimension	60
out_dimension	180
in_directory	/home2/st119799/OssianFinal/.../cmp//
lf0	
in_dimension	1
out_dimension	3
in_directory	/home2/st119799/OssianFinal/.../cmp//
vuv	
in_dimension	0
out_dimension	1
in_directory	
bap	
in_dimension	5
out_dimension	15
in_directory	/home2/st119799/OssianFinal/.../cmp//
multistream dimensions	[199]
use_rprop	0
dropout_rate	0.0

Continuation of Table A.2	
Parameters	Values
projection_outsize	10
batch_size	256
model_type	DNN
hidden_layer_size	[1024, 1024, 1024, 1024, 1024, 1024]
projection_learning_rate_scaling	1.0
pretraining_epochs	10
initial_projection_distrib	gaussian
index_to_project	0
l2_reg	0.0
warmup_momentum	0.0
training_epochs	50
hidden_activation	tanh
hidden_layer_type	['TANH', 'TANH', 'TANH', 'TANH', 'TANH', 'TANH']
sequential_training	True
do_pretraining	False
projection_insize	10000
l1_reg	0.0
output_activation	linear
pretraining_lr	0.0001
momentum	0.9

Table A.3 shows the performance measurement of developed biopsy diagnostic models .

Table A.1: A typical linguistic and prosodic HTS context for English language. Reprinted from (Zen, 2006).

p_1	the phoneme identity before the previous phoneme
p_2	the previous phoneme identity
p_3	the current phoneme identity
p_4	the next phoneme identity
p_5	the phoneme after the next phoneme identity
p_6	position of the current phoneme identity in the current syllable (forward)
p_7	position of the current phoneme identity in the current syllable (backward)
a_1	whether the previous syllable stressed or not (0: not stressed, 1: stressed)
a_2	whether the previous syllable accented or not (0: not accented, 1: accented)
a_3	the number of phonemes in the previous syllable
b_1	whether the current syllable stressed or not (0: not stressed, 1: stressed)
b_2	whether the current syllable accented or not (0: not accented, 1: accented)
b_3	the number of phonemes in the current syllable
b_4	position of the current syllable in the current word (forward)
b_5	position of the current syllable in the current word (backward)
b_6	position of the current syllable in the current phrase (forward)
b_7	position of the current syllable in the current phrase (backward)
b_8	the number of stressed syllables before the current syllable in the current phrase
b_9	the number of stressed syllables after the current syllable in the current phrase
b_{10}	the number of accented syllables before the current syllable in the current phrase
b_{11}	the number of accented syllables after the current syllable in the current phrase
b_{12}	the number of syllables from the previous stressed syllable to the current syllable
b_{13}	the number of syllables from the current syllable to the next stressed syllable
b_{14}	the number of syllables from the previous accented syllable to the current syllable
b_{15}	the number of syllables from the current syllable to the next accented syllable
b_{16}	name of the vowel of the current syllable
c_1	whether the next syllable stressed or not (0: not stressed, 1: stressed)
c_2	whether the next syllable accented or not (0: not accented, 1: accented)
c_3	the number of phonemes in the next syllable
d_1	gpos (guess part-of-speech) of the previous word
d_2	the number of syllables in the previous word
e_1	gpos (guess part-of-speech) of the current word
e_2	the number of syllables in the current word
e_3	position of the current word in the current phrase (forward)
e_4	position of the current word in the current phrase (backward)
e_5	the number of content words before the current word in the current phrase
e_6	the number of content words after the current word in the current phrase
e_7	the number of words from the previous content word to the current word
e_8	the number of words from the current word to the next content word
f_1	gpos (guess part-of-speech) of the next word
f_2	the number of syllables in the next word
g_1	the number of syllables in the previous phrase
g_2	the number of words in the previous phrase
h_1	the number of syllables in the current phrase
h_2	the number of words in the current phrase
h_3	position of the current phrase in utterence (forward)
h_4	position of the current phrase in utterence (backward)
h_5	TOBI endtone of the current phrase
i_1	the number of syllables in the next phrase
i_2	the number of words in the next phrase
j_1	the number of syllables in this utterence
j_2	the number of words in this utternce
j_3	the number of phrases in this utterence

Table A.3: Parameters of RNN architecture based duration model

Parameters	Values
in_dimension	5
out_dimension	5
multistream dimensions	[5]
use_rprop	0
dropout_rate	0.0
projection_outsize	10
early_stop_epochs	5
warmup_epoch	1000
learning_rate	0.0001
layers_with_projection_input	[0]
batch_size	256
model_type	DNN
hidden_layer_size	[512, 512, 512]
projection_learning_rate_scaling	1.0
pretraining_epochs	10
initial_projection_distrib	gaussian
index_to_project	0
l2_reg	0.0
warmup_momentum	0.0
training_epochs	200
hidden_activation	tanh
hidden_layer_type	['TANH', 'TANH', 'TANH']
sequential_training	True
do_pretraining	False
projection_insize	10000

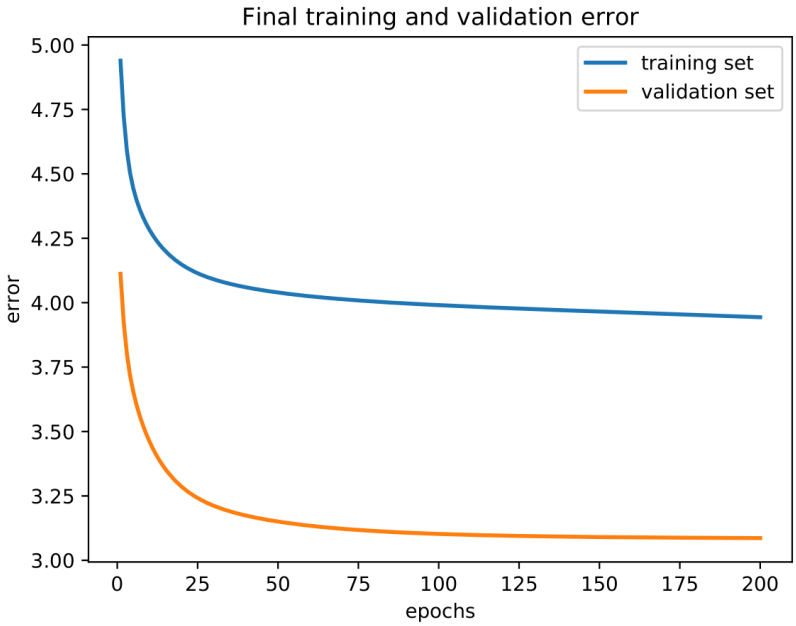


Figure A.1: Convergence plot of Wolwala 0.1 duration model.

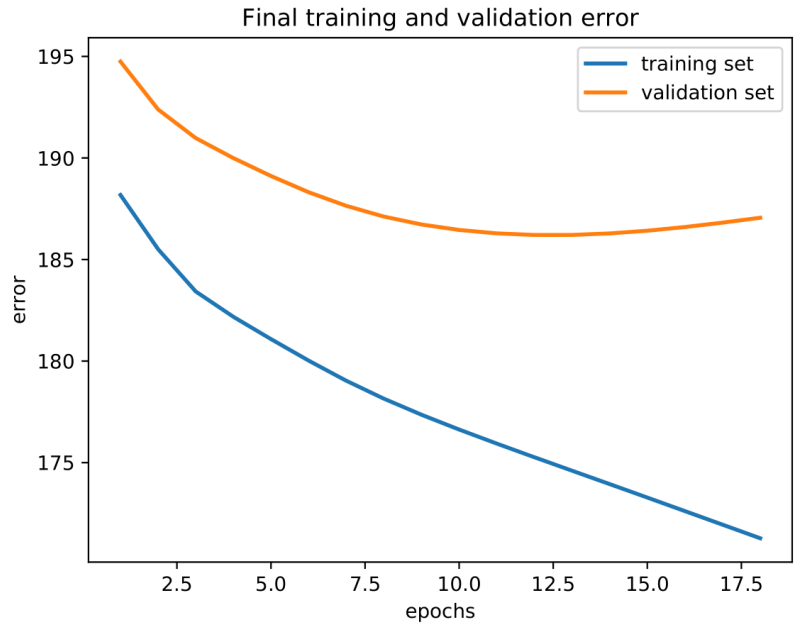


Figure A.2: Convergence plot of Wolwala 0.1 acoustic model.

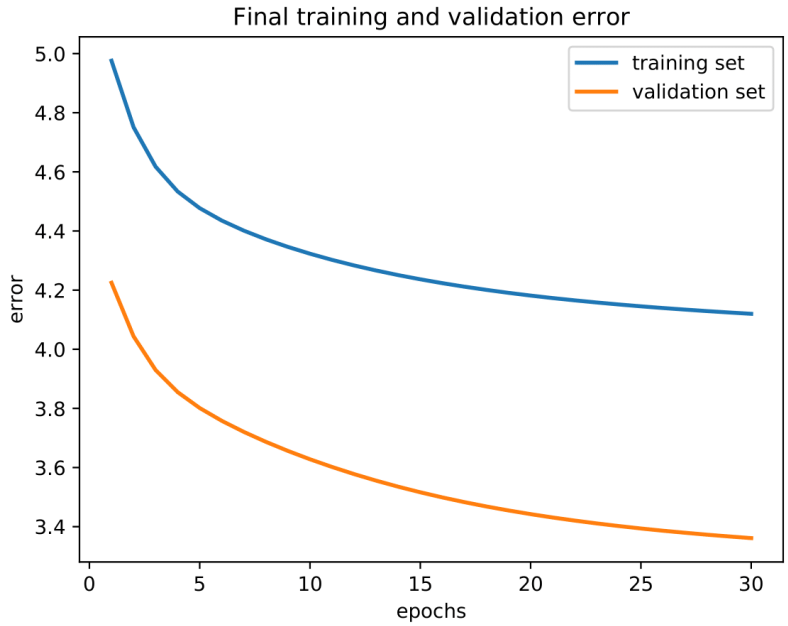


Figure A.3: Convergence plot of Wolwala 0.2 duration model.

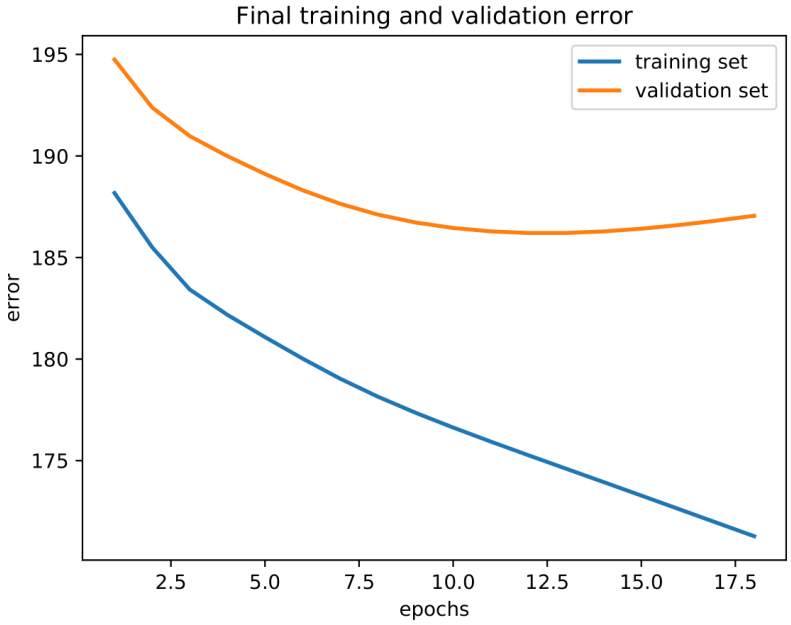


Figure A.4: Convergence plot of Wolwala 0.2 acoustic model.

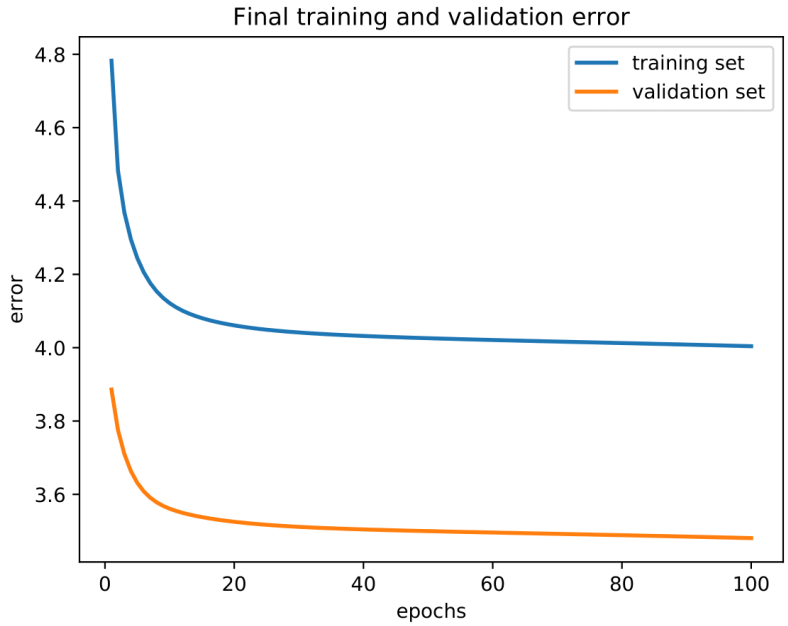


Figure A.5: Convergence plot of Wolwala 0.3 duration model.

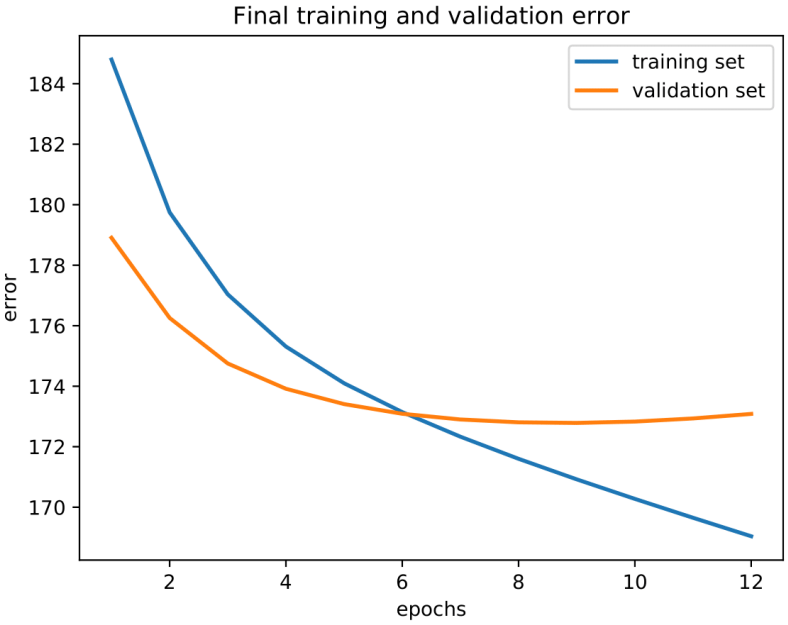


Figure A.6: Convergence plot of Wolwala 0.3 acoustic model.

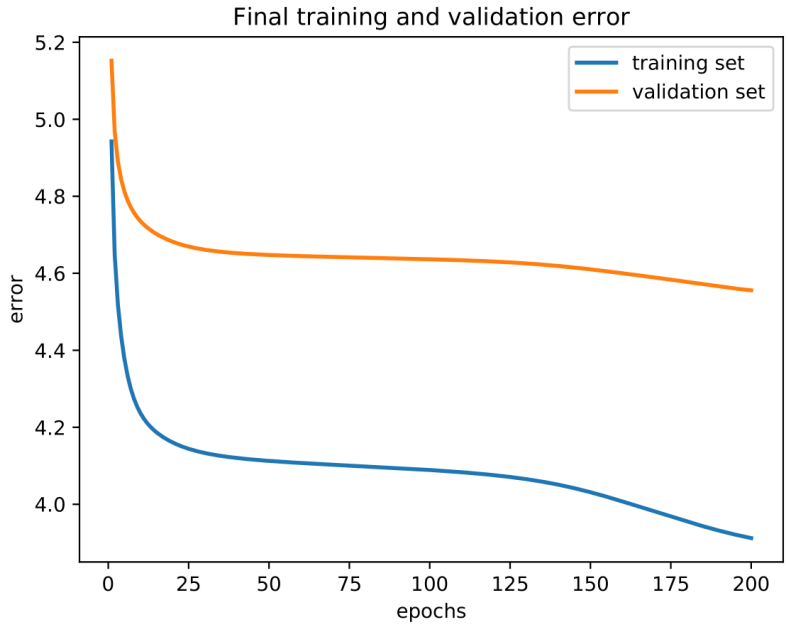


Figure A.7: Convergence plot of Wolwala 0.4 duration model.

Figure A.8: Convergence plot of Wolwala 0.4 acoustic model.