TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

THESIS NO: 075MSCSK003

Attention And WaveNet Vocoder Based Nepali Text-To-Speech Synthesis

by

Ashok Basnet

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SYSTEM AND KNOWLEDGE ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

AUGUST, 2021

Attention And WaveNet Vocoder Based Nepali Text-To-Speech Synthesis

by
Ashok Basnet
075MSCSK003

Thesis Supervisor
Asst. Prof. Dr. Basanta Joshi

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer System and Knowledge Engineering

Department of Electronics and Computer Engineering
Institute of Engineering, Pulchowk Campus
Tribhuvan University
Lalitpur, Nepal

August, 2021.

COPYRIGHT©

Head
Department of Electronics and Computer Engineering
Institute of Engineering, Pulchowk Campus
Pulchowk, Lalitpur, Nepal

DECLARATION

I declare that the work hereby submitted for Master of Science in Information and Communication Engineering (MSICE) at IOE, Pulchowk Campus entitled **"Attention And WaveNet Vocoder Based Nepali Text-To-Speech Synthesis"** is my own work and has not been previously submitted by me at any university for any academic award.

I authorize IOE, Pulchowk Campus to lend this thesis to other institution or individuals for the purpose of scholarly research.

Student Name: Ashok Basnet
Roll No.: 075MSCSK003
Date: August, 2021.

RECOMMENDATION

The undersigned certify that they have read and recommended to the Department of Electronics and Computer Engineering for acceptance, a thesis entitled **"Attention And WaveNet Vocoder Based Nepali Text-To-Speech Synthesis"**, submitted by **Ashok Basnet** in partial fulfillment of the requirement for the award of the degree of **"Master of Science in Computer System and Knowledge Engineering"**.

....

Supervisor:
Asst. Prof. Dr. Basanta Joshi,
Department of Electronics and Computer Engineering,
Institute of Engineering, Tribhuvan University.

....

External Examiner:
Assoc. Prof. Dr. Bal Krishna Bal,
Head of Department,
Lead Researcher: Information and Language Processing Research Lab,
Department of Computer Science and Engineering,
School of Engineering, Kathmandu University.

....

Committee Chairperson:
Assoc. Prof. Dr. Nanda Bikram Adhikari,
Department of Electronics and Computer Engineering,
Institute of Engineering, Tribhuvan University.

Date: August, 2021.

DEPARTMENTAL ACCEPTANCE

The thesis entitled **"Attention And WaveNet Vocoder Based Nepali Text-To-Speech Synthesis"**, submitted by **Ashok Basnet** in partial fulfillment of the requirement for the award of the degree of **"Master of Science in Computer System and Knowledge Engineering"** has been accepted as a bonafide record of work independently carried out by him in the department.

....

Prof. Dr. Ram Krishna Maharjan,
Head of the Department,
Department of Electronics and Computer Engineering,
Pulchowk Campus,
Institute of Engineering,
Tribhuvan University,
Nepal.

ACKNOWLEDGEMENT

With immense pleasure, I would like to express my sincere gratitude to my thesis supervisor **Asst. Prof. Dr. Basanta Joshi** for his invaluable insights and support. I felt very lucky that my supervisor organized seminar in the domain of Deep Learning Based Text and Speech, which helped me a lot during the thesis work. Due to his continuous support and guidance from the thesis idea formulation to the completion, I am now able to put forward my work.

My sincere thanks to our program coordinator **Assoc. Prof. Dr. Nanda Bikram Adhikari** for providing us with the necessary instructions and guidelines of thesis. I am also grateful to all the teachers of the Department for their constructive comments to shape this work. I am also indebted to Suman Sharma and Bharat Bhatta for their creative and helpful suggestions. They spent a lot of their valuable time guiding and supporting me in all the time of research and writing of this thesis.

Nevertheless, I am also thankful to Yashasvi Raj Pant, Rupesh Shrestha, Uddav Khanal, Bharat Bhatta, Prativa Paudel, Barsha Sitoula, Vrishav Khanal, Mukesh Rai, Sagar Rai and Aashish Karki for their contribution during qualitative evaluation of model.

Finally, I would also like to express my hearty gratitude to the **Department of Electronics and Computer Engineering**, Pulchowk Campus along with my family and friends for their continuous suggestion and support.

Ashok Basnet
075MSCSK003

ABSTRACT

Since the evolution of Artificial Intelligence, researchers in the field of audio ai are constantly trying to figure out the way for making text-to-speech systems more naturally resonating and are directed towards constructing the human level of voice synthesis network. Synthesis of spoken language from the written text is the major objective of Text-to-Speech synthesis. Such network has a vibrant scope in the field of human-computer inter linkage. Research on deep learning has shown the possibility to infer near human level natural speech from the input text. This work presents the idea for developing end-to-end Nepali speech synthesis network using encoder-decoder architecture conditioned on attention mechanisms followed by WaveNet as Vocoder. The RNN based seq-to-seq feature prediction deep network maps the input character embedding into the latent space representation which is decoded into mel-spectrogram representation. Mel-spectrogram is then converted into the audio waveform by WaveNet vocoder model trained for synthesizing the human speech. The main challenges of the work is the need of high computational power and large data of high quality transcribed audio. Here the network is trained on the Nepali speech dataset from OpenSLR having 157,000 utterances of 165 hours from 527 speakers. The synthesized speech is clear in quality and can be understood by the listener. The quality of synthesized speech was evaluated by listening (i.e. by Mean Opinion Score test). The synthesized sample of speech attained MOS of 3.07, when 40 samples subjected to 10 volunteers. The deep neural network can be trained directly from the data without relying on complex feature engineering, and achieves an acceptable audio quality.

Keywords: Text-to-Speech synthesis, Attention Mechanism, WaveNet, Recurrent Neural Network (RNN), Seq-to-Seq, Vocoder

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

TTS	Text to Speech
ACTA	Assistive Context-aware Toolkit
MOS	Mean Opinion Score
Seq2Seq	Sequence to Sequence
CNN	Convolution Neural Network
DNN	Deep Neural Network
OOV	Out of Vocabulary
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
DCTTS	Deep Convolutional Text To Speech
ReLU	Rectified Linear Unit
GAN	Generative Adversarial Network
NC Conv.	Non causal Convolution
STFT	Short Time Fourier Transform
FFT	Fast Fourier Transform
GPU	Graphics Processing Unit
TPU	Tensor Processing Unit
OpenSLR	Open Speech and Language Resources
MOLD	Mixture of Logistic Distributions
BPTT	Backpropagation Through Time

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Many of us have seen the great scientist Stephen Hawking on his chair with a ACTA (assistive context-aware toolkit) computer. This computer helped him to deliver his lecture. It was a traditional form of TTS synthesizer developed by Intel. Traditional Text to Speech models were composed of many domain specific modules such as a text analyser, Fo generator, mel spectrum generator, pause estimator along with vocoder for synthesizing audio waveform from the data. These modules were complicated to build and maintain and also the error in each module is inherited to a consecutive model resulting in accumulation of noise and error. Other two traditional methods i.e. concatenative approach where speech stored in a large database was used to generate new audio signal and parametric approach which contains function with sets of parameters for modifying speech were extinct because of their inability to properly maintain the prosody of the speaker often producing robotic sound. Well, after the release of google's WaveNet [1], the era of natural sounding speech generation has begun. Since then, research direction towards the data efficient, more natural sounding, end to end TTS system has begun. WaveNet is an auto-regressive and fully probabilistic model for predicting the distribution for speech signals. It consists of convolutional units instead of the recurrent, so it has more computational efficiency. Also the model is able to to maintain long range temporal dependencies which is fundamental requirements for natural sounding audio waveform generation.

In Nepali text to speech work trained on Flite TTS engine [2] based on Carnegie Mellon University, the synthesized sound is very robotic and noisy. In addition to that, Festival system implemented for building a Nepali voice from the text displayed on the screen possess issues such as overlaps and echoes in the synthesized speech[3]. The Tacotron TTS [4] was developed using an encoder, attention based decoder with a post processing unit. Griffin-lim algorithm is used to generate the waveform from the output spectrogram of the tacotron network. Deep voice 1 [5], Deep voice 2 [6], Deep voice 3 [7], Voice loop [8] were other TTS models that also developed synthesized speech. But, Tacotron 2 [9] is viewed as one of the most successful models developed yet with a MOS of 4.53 for meridian english as claimed by Google. The combined setup of Recurrent seq2seq feature prediction network for mapping the character embedding to the mel spectrogram followed by WaveNet's model which synthesizes the audio waveform from the mel-scale spectrogram forms the Tacotron 2. Such end-to-end systems are able to infer the speech waveform from the input text without the hand crafted feature engineering like the previously developed model. In the era of Deep Learning, end-to-end neural networks, which are the state of the art for speech recognition tasks, have become highly competitive with the conventional TTS systems.

Speech Synthesis is called the artificial production of human speech. Its aim is to synthesize intelligible and natural audio which is indistinguishable from human recordings. A speech synthesizer is a computer system used for this purpose.The network works by combined effort of two stage neural networks. First part is conversion of the textual information into linguistic specification by the use of Tacotron-2 architecture. Second part of this work is based on WaveNet architecture which is able to convert the linguistic specification into the speech waveform. Implementing Attention Based Recurrent Sequence to Sequence model with WaveNet vocoder for building the Nepali TTS model appears as optimistic candidates to improve the performance. Here every module can be trained at once such that global performance criterion is optimized.

1.2 Problem Definition

Democratizing AI is one of the fundamental slogans in the AI community. But still there exist biases in the available resources. Tacotron 2 by Google claims MOS of 4.53 where MOS of 16 bit PCM recorded speech is 4.58 in English [9]. It has shown near human level performance for English language. Similar results can be seen for Chinese TTS.

But, only a little work has been done on the Nepali language and appears far less sufficient to produce natural results. The previous work on the Nepali TTS seems to produce foggy, robotic and noisy speech [2]. This seems to appear due to insufficient noise free spoken corpus in the Nepali language. Also the other work on Nepali Text-to-Speech synthesis suggest issue of insufficient linguistic resources for under-resourced languages like Nepali [10]. There has been much work on the English language. So, this work primarily focuses on preprocessing of the Nepali Speech dataset from OpenSLR [11] [12]. Secondly it will develop the Nepali TTS model using Attention Based Recurrent Sequence to Sequence model with WaveNet vocoder and finally evaluate its performance.

1.3 Objectives

The Primary objectives to conduct this research work are:

- To build and train an attention based Mel-spectrogram prediction network from input character.
- To develop a base model for Nepali TTS synthesis using Neural WaveNet Vocoder.

1.4 Thesis Contribution

This work is an end to end system for Nepali Text-to-Speech synthesis. The work distinctly able to contribute in the following two ideas:

- The model trained for Nepali Text-to-Speech synthesis can be implemented for various task such as automatic news reading, audio book preparation and so on. Hence, it plays prime role in TTS applications.
- Previously developed model for TTS implemented Griffin-lim algorithm to vocode audio waveform from the mel-spectrogram. This works has shown that WaveNet can be used to infer more clean audio waveform than that of previously implemented methods.

1.5 Research Challenges

During this research the major challenges faced are:

- **Text Normalization:** Properly formatted Nepali transcribed text is crucial for clean speech generation. But, it is a great challenge to normalize numbers, dates, telephone numbers, percentages, emails, URLs, addresses and so on.
- **Computational Resources:** The total audio size used for training was about the size of 9GB and of 165 hours of audio clips. It is a great challenge to handle such large computationally intensive data with a less powerful GPU/TPU. Also, the need for number of epochs and iterations for training the network is higher in order to properly stabilize the loss of the model.
- **Hyper-parameter Tuning:** Very large set of hyper-parameters are associated in this experiment and tuning every parameter and if less computational power is available, retraining the network and fine tuning every parameter might take years of effort.

CHAPTER 2

LITERATURE REVIEW

Parametric and Concatenative speech synthesis were popular methods for inferencing speech in past decades. But these methods were resource intensive, required manpower for feature engineering and had complex pipelines. The features were mostly language specific. All the work from scratch needs to done in order to adopt the system for a new language. Hence they were very language specific.

2.1 Previous Approach in Speech Synthesis

In an **exemplar-based** voice synthesis network, the labeled speech corpus are stored in a database such that relevant parts can be searched during the inference phase. Here, the query is performed on a large database of stored speech and certain portions of the speech are selected on the basis of how well the specification is matched. The specification and the units are completely described by a feature structure, which can be any mixture of linguistic and acoustic features. Here the speech is directly derived from the recorded voice samples and it appears that the larger the database, the better the coverage. Using this system one can only derive the speech which are stored in the database.[13]

Speech synthesis via **Statistical Parameter** obtained attention in the AI community as it was not using the stored examples. This model uses parameters such as mean, variance of PDF for capturing the parameters of the training data. It can transform voice characteristics, speaking styles and emotions by transforming its model parameters. It has multilingual support. But the quality of speech was very low. Three factors were supposed to degrade the quality i.e. vocoder, over smoothing during speech inference and the modelling accuracy. Researcher struggled to improve performance of vocoder. On the other hand, the invention of more advanced statistical models, such as the trajectory HMMs, leaded to smoother speech trajectories and increased user opinion scores.[14]

Another powerful model **Hidden Markov Models** was also unable to produce high quality speech because there are such assumptions which could not be implemented in the speech [15]. In HMM a parametric model is involved which degrades the final audio quality. No matter how naturally the models generate parameters, the final quality is very much dependent on the model used. The model is also unable to generate delicate phenomena in waveform such as prosody of the speaker. Also another main bottleneck in HMM is that, it assumes state of model t-1 determines the next state which doesn't work for voice as speech requires long term dependencies. However this approach have been implemented for Nepali ASR [16].

With the aim to correctly synthesizing the natural speech from Nepali words or sentences, the Nepali text-to-speech synthesis system is developed by using the concatenative approach employing Epoch Synchronous Non-Overlap Add Method (ESNOLA) in 2013. This approach uses a signal dictionaries having raw sound signal representing parts of phonemes as a speech database. The database retrieve relevant parts of phonemes on the basis of query performed by input text. The pitch of the previously recorded speech signal remains intact by this action while taking care of aspects such as personality, naturalness, quality assessments and platform independence [17].

Developing high quality text-to-speech for underresourced languages like Nepali possess huge challenges because of scarcity of high quality linguistic resources. Preparation of such linguistic resources is time consuming, costly and demands the involvement of the linguistic experts. Work on Nepali TTS [18] by concatenative approach appears to require a lot of feature engineering. In 2017, modification on the existing Nepali TTS [19] was performed by adding the post and pre-processing modules. The system was made public as desktop application and as well as plugin for Firefox [10]. Qualitative Evaluation by MOS of this modified TTS with [10] with existing system [**?**] has shown an overall improvement of 6 percentage in terms of naturalness and intelligibility, whereas the result of comprehension and diagnostic rhyme test is increased by 12 percentage and 10 percentage respectively. 30 users were involved in the evaluation and were asked to provide their evaluation of the systems on the basis of parameters-intelligibility and naturalness.

2.2 Text Processing

Text-to-speech system is basically composed of two stages, first is intermediate representation from input text and second consists of waveform inference from intermediate representation. The major focus in text normalization is to replace non-standard input text with system supported linguistic textual representation. The transcribed text can be classified in respect to the semiotic type and the unstructured text could be transformed to structured and unambiguous representation. The basic steps for text normalization involves: the identification of text genre, splitting of sentence, Tokenization, token classification into semiotic classes, decoding to find underlying token, verbalisation of non natural language, , homograph resolution and parsing. Mainly there are three class of linguistic ambiguity involved during speech synthesis in TTS: semantic, grammatical and word identity and this needs to be addressed via text decoding. [15]

During text preprocessing, it is necessary to normalize the transcribed text. **Text Normalization** involves conversion of upper case to lower case (i.e. applicable for roman script, hence script dependent), spelling out the numbers and standardizing non-standard text. Non-standard words include dates, abbreviations, acronyms, symbols, currency and numbers. These Non-standard words greatly impact on the model performance and create pronunciation issues on synthesized speech. The identified semiotic classes are: ordinal numbers, cardinal numbers, years, dates, telephone numbers, money, emails, measures, percentage, urls, real estate, computer programs, addresses and abbreviations. Typical technology for text normalization involves sets of ad hoc rules tuned to handle genres of text with the expected result that the techniques do not usually generalize well to new domains. [20]

2.3 Character Embedding

The major purpose of character vectorization is to represent any segment of sentences into its feature set such that it can be used for model training. The embedding process transforms the input text into the set of vector representations. Such vectors have the capability of capturing the semantic meaning of words and syntax i.e. grammatical function. This enables us to perform mathematical operations on them. In word level embedding there is the possibility of encountering OOV words. This confuses our model so there is need of another embedding mechanism known as character level embedding to handle such confusion due to OOV. Character level embedding looks at the character level composition and uses 1D CNN for finding numeric representation. 1D CNN provides the capability of information elicitation from short segments of long transcription. Character embedding possesses two advantages over word level embedding. First, it could handle misspellings and slang words. Secondly, the matrix size required for embedding gets smaller than that of word level. [21]

2.4 Sequence-to-Sequence models

Deep Neural Networks are very potent machine learning models that achieve outstanding performance on difficult problems such as speech recognition and visual object detection. Supervised learning in Deep Neural Network can be trained with backpropagation to solve complex problems such as object detection, speaker recognition. For these inputs parameters and targeted output are encoded into a fixed dimension vector representation. It is a bottleneck for Deep Neural Networks. Such network could not deal with the problems as machine translation, question answering system, speech recognition and other sequential problems. In such class of problem one could not know the sequence length in prior. This demands domain independent procedure which is capable of mapping the sequence of input units to the sequence of targeted output.

Researcher found that implementation of LSTM architecture could solve Seq-to-Seq problems. Basically two LSTM are used. First one reads the input sequence one time-step at one time. This sequence is converted to large vector representation of fixed dimension. Second LSTM network is used to extract sequence of output from a fixed dimension vector. Seq-to-Seq model used with attention networks appears to give state of art models in many DNN applications. [22]

RNN Encoder-Decoder: This Encoder-decoder network is for encoding the variable length input sequence to the fixed length vector representation followed by decoder for decoding this fixed length vector into output sequence of variable length. Thus the model is able to learn the conditional distribution between two variable length sequences.

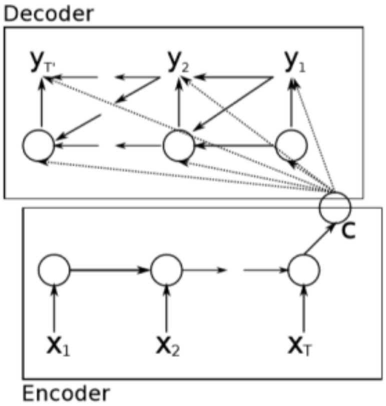


Figure 2.1: RNN Encoder/Decoder Architecture [23].

2.5 Attention Network

Alignment Mechanism : The main disadvantage of the sequence to sequence model is that it produces a fixed length context vector. Bahdanau successfully implemented an attention mechanism to overcome the problem of necessity to represent input sequence by fixed length vector representation in encoder-decoder architecture. The attention mechanisms are frequently used in transduction models, sequence modeling and so on. It allows to model dependencies without regard to the input and output sequence.This context vector is incapable of remembering the sequence once it has processed the sequence. So to overcome this inefficiency a neural network is introduced in between encoder and decoder which serves the attention mechanism. Attention allows the model to focus on the relevant parts of the input sequences as needed. It differs from seq2seq in a way that its encoder passes more data to the decoder. In case of seq2seq only the last hidden state of the encoding stage is passed whereas it passes all the hidden state to the decoder. To focus on certain parts of input it looks at a set of hidden states to find the most associated word. Each hidden state is given a score and finally each hidden state is multiplied by its soft maxed score. The reason for using softmax is to amplify the high score hidden state and drown out the low score state. This frees a neural translation model from having to squash all the information of a source sentence, regardless of its length, into a fixed - length vector.Hence, this equips network to deal with long sentences.[23]

Let's say y_t is the generated sequence of words during translation, then soft-searches looks for the position (1, ..., T) in input sequence x = (x_1, ..., x_T) to find out where the most matching information are concentrated. Hence the word y_t predicted by decoder is dependent on context vectors that relate to current source position as well as on previously generated words s_{t-1}, y_{t-1}.

$$c_t = \sum_{j=1}^{T_x} \alpha_{tj}.h_j \tag{2.1}$$

Following operations are carried out in the decoder:

- **Computing Alignment:**

  $$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=i}^{T_x} exp(e_{ik})} \tag{2.2}$$

  Where, j is the input position and i is the output position.

- **Computing e_{ij} using feed-forward Network:**

  $$e_{ij} = v^T.tanh(Ws_{i-1} + Vh_j) \tag{2.3}$$

  Where v, V and W are weights for training networks.

- **Calculating Context:**

  $$c_t = \sum_{j=1}^{T_x} \alpha_{tj}.h_j \tag{2.4}$$

- **Updated Decoder State:**

  $$s_i = f(s_{i-1}, y_{i-1}, c_i \tag{2.5}$$

- **Predicting new output:**

$$argmaxp(y_i/y1, ..., y_{i-1}, x = g(y_{i-1}, s_i, c_i) \tag{2.6}$$

This mechanism frees the encoder from the burden of encoding input sequence into fixed length context vector representation. This also helps in spreading the information throughout the annotations sequence. Hence, decoders retrieve information on the basis of priority or importance of sequence.

Attention Mechanism: The ability to focus on a specific subset of inputs in a neural network can be added using attention. There are two classes of attention i.e. soft and hard attention. In soft attention, features are mapped between zeros and ones whereas in hard attention the values are constrained to be either zero or one. [24]

Attention Formalization: During machine translation, long sequences of sentences at inputs disable the model to remember the whole sequence. Attention takes samples from the chunk of long sequence and use it for translation. Hence, attention is used for computing the affinity of encoder states with the decoder states. Suppose we have n encoder states. Say a_i represents the affinity of encoder state i with the decoder state. Say there exists a $h_{1:n}$ encoder state and s_{t-1} is a decoder state. Then,

$$\alpha_i = f(h_i, s_{t-1}) = h_i T s_{t-1} \tag{2.7}$$

$$a = softmax(a) \tag{2.8}$$

$$c = \sum_{i=1:n} h_i a_i \tag{2.9}$$

2.6 Tacotron

Tacatron, developed by google also known as end to end speech synthesis model, achieved 3.82 mean opinion score on US english for TTS. This model can be trained by random initialization from scratch given the raw <text, audio>. This network is composed of recurrent seq-to-seq feature predictions. The network maps the character embeddings into the mel spectrogram. It is often difficult to deal with the large variations in pronunciation of the same text in different words and when spoken by different people. This problem can be dealt with using the attention paradigm in sequence to sequence model and the method doesn't require phoneme-level alignment. The high level intuition of this method is that it takes characters as the raw input, and produces spectrogram frames which are transformed into the audio waveforms. In order to vocode the spectrograms, Griffin-Lim Algorithm followed by STFT is used. Griffin-Lim Algorithm is used for phase estimation. [4]

Tacotron 2 followed seq-to-seq approaches for text to mel as that of Tacotron but uses WaveNet as vocoder for synthesizing the audio signal from the spectrograms. The architecture is divided into two parts, first is the attention based recurrent seq2seq feature prediction network on the basis of which mel spectrograms frames are derived from the text and second is a WaveNet that generates the audio waveform from the predicted spectrogram frames. Tacatron is said to produce very promising results with an MOS of 4.526 but the need for data volume is very large with high computational power. [9]

Intermediate features representation: Mel-scale frequency spectrogram is used in tacotron as a representation bridge between two components. This representation can easily be computed from the time domain waveforms. Hence, this intermediate feature enables the training of the two components of the network separately. As mel-spectrogram is phase invariant, it enables easier implementation of backpropagation of squared error loss. Otherwise, it would be difficult to work on the waveform. Short Time Fourier Transform also known as linear frequency spectrogram and this mel frequency spectrogram are related. mel-spectrogram is actually inspired by the human auditory system. It enables us to represent frequency content in fewer dimensions.

2.7 Neural Vocoder

Those networks which are capable of deriving audio waveforms from the given audio features are known as Neural Vocoder. Two vocoders are mostly used in TTS synthesis. One is the Griffin-Lim Algorithm and the other is modified WaveNet. WaveNet is an autoregressive system. It can generate a very natural sound if conditioned properly. It is a statistical vocoder. But training the WaveNet takes a longer time. Hence, most of the TTS uses Griffin-Lim algorithm during the training.

Griffin-Lim algorithm serves as an algorithm for recovering the timing of an audio signal provided the magnitude or the spectrogram. This algorithm works iteratively to recover the magnitude and the algorithm usually converges in about 40 iterations. It has lower performance than that of wavenet but is efficient for spectrogram inversion during debugging of the new system. [25]

Algorithm 1 Griffin-Lim algorithm

```
procedure INPUT: $s$ SPECTROGRAM (())
    for $i$ = 1 to n-iterations do
        $c1$ = STFT(x)                          ▷ $x$ = random($n$ samples)
        $a$ = angle($c_1$)
        $c_2$ = $s$. $e^{ja}$
        $x$ = ISFTF($c_2$)
    end for
end procedure
```

2.8 WaveNet

WaveNet is an autoregressive model which has the capability to predict the probability distribution of the next sample, given the previous samples and an input conditioning signal. It produces a whole sequence of samples by feeding a model with previously generated samples. Choice of softmax as probability distribution in WaveNet makes both training and synthesis tasks computationally tractable. Conditioning of WaveNet on acoustic features convey prosodic and verbal information. These variables are upsampled using the desired frequency and those are fed to the network of WaveNet via the conditioning network. WaveNet is made up of two modules, first is a convolution stack and second is post processing module.

The convolution stack is composed of dilated convolution residual units. This unit performs multi scale feature extraction. Post-processing unit simply combines the output information from these residual blocks to infer the next sample. WaveNet introduces a new kind of generative model that operates directly on raw audio waveforms. Say x= (x_1 , x_2 , ..., x_T) be the joint probability of the waveform then a product of conditional probabilities is given as:

$$p(x) = \prod_{t=1}^{T} p(x_t|x_1, ..., x_{t-1}) \tag{2.10}$$

Every audio sample x_t is conditioned on the sample at previous timesteps. Hence the main part of the wavenet is causal convolutions. Use of causal convolutions reduces the training time of the network. Stack of convolutional layers is used to model conditional probability distributions. Pooling mechanism is not implemented, hence the time dimensionality of output remains unchanged to that of input. Dilated convolution can be used to widen/increase the receptive field as dilation skips the values involved in convolution. It is analogous to pooling but the input dimensions equals the output dimensions. This convolution technique is popular on signal processing tasks as well as on image segmentation. This concept of WaveNet can be implemented for music audio modeling, multi-speaker speech generation tasks as well as on Text To Speech systems. Such generated output is considered as of highest naturalness, and is applied to a wide range of tasks. [1]

The residual block in WaveNet consists of expert and gate. Element-wise multiplication is performed on the output of these two units.

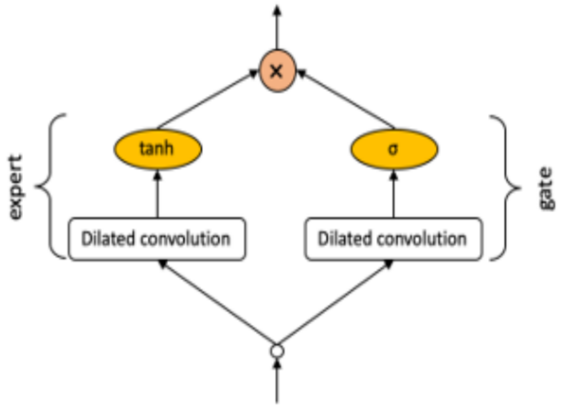


Figure 2.2: Stack of residual block in WaveNet [1].

For any block i, hidden state vector ($z^{(i)}$) can be computed using:

$$z^{(i)} = tanh(W_f^{(i)} * x^{(i-1)} + L_f^{(i)}) \odot \sigma(W_g^{(i)} * x^{(i-1)} + L_g^{(i)}) \tag{2.11}$$

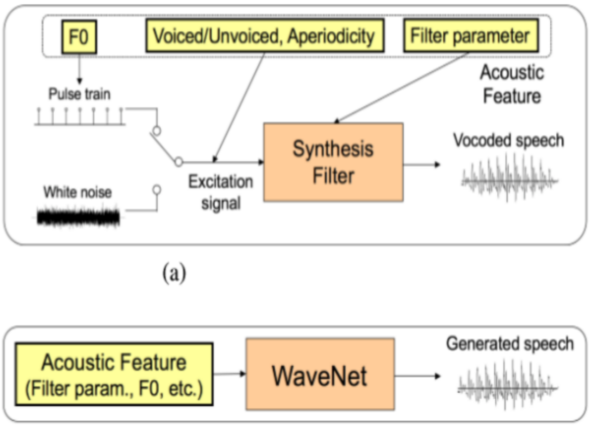
Where $x^{(i-1)}$ is input and $x^{(i)}$ is output.

$\odot$ denotes element wise multiplication and $*$ denotes convolution.

Although the developed Deep Neural Networks (DNN) were powerful enough to learn difficult non linear tasks, it could not map seq2seq. So, multi-layer LSTM (Long Short Term Memory) was used to produce fixed dimension vectors from input sequence and an additional LSTM can be used to decode the vector to obtain the targeted sequence. This has a huge advantage on tasks like language translation. The ability of LSTM to learn long range temporal dependencies made it a natural candidate for the application.

2.9 WaveNet Vocoder

The purpose of WaveNet Vocoder in TTS is to invert mel-spectrogram into audio waveform.



(a)

Figure 2.3: Vocoder : a. conventional vocoder b. WaveNet vocoder [26].

WaveNet vocoder showed that it is possible to add features like breathing while talking, pauses in speech, mouth movement and other speech features to infer more natural speech. Hence, WaveNet vocoder has got lot of attention to obtain state-of-the-art result in TTS. [26]

CHAPTER 3

RELATED THEORY

3.1 Components of Neural Network

RNN: RNN was first proposed in the 1980s but its potential has only been realized after the availability of massive computational power. In contrast to feed forward deep neural networks, it uses its memory (i.e. internal state) for processing input data. Internal state of RNN is capable of remembering the important parameters in input data. This feature makes RNN capable of predicting the next event. Hence, used in financial data, time series, text, speech and so on. For making a decision, it considers the current input as well as its previous state. The information decays in an exponential manner with respect to time.

Backpropagation Through Time: In high level overview, BPTT propagates the error from the output layers towards the input layer (i.e. from last timestep to the first open). In the case of RNN, BPPT is done in an unrolled network. The error at a particular time step is dependent on the error of the previous timestep. The calculated error is used for updating weights of the neuron such that error is minimized. The bottleneck for BPTT is that its computational cost is high for a higher number of timesteps.

Truncated BPTT: It is another popular method used in training the recurrent network. It is computationally less intensive than BPTT. The accumulation of gradients in the network result in network instability. Hence network becomes passive in learning. The weight might become so large that, exploding gradients problems arise. Truncated BPTT can solve such issues.[27]

LSTMs: Long Short-Term Memory: To suffice the incapability of RNN to learn long term dependencies, LSTM a network was developed. Extended memory was added in RNN to develop LSTM. Such a memory cell consists of linear units and logistic units with multiplicative interactions. LSTMs added capability of remembering input over a longer period of time in RNN. Information in memory cells can be deleted, appended and read. The gated cells in memory are responsible for this decision making. LSTM consists of three gates i.e. input gate, forget gate and output gate. The gated cell learns the priority of information over time and later such learned weights is used to rank the priority of information. On the basis of these priorities, the gate decides whether to store or forget information.

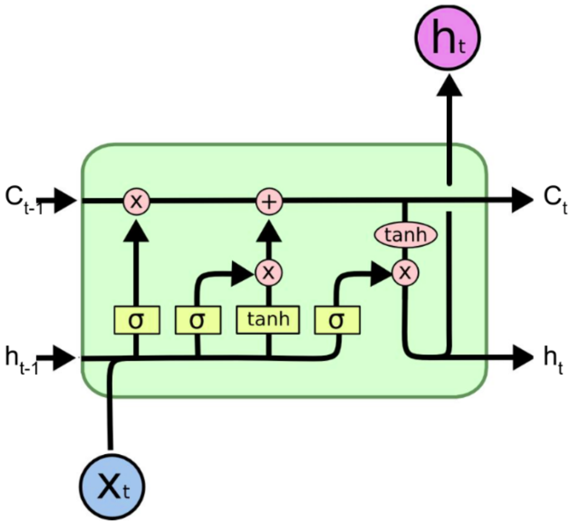


Figure 3.1: Block Diagram of Long-Short-Term-Memory [28].

Forget Gate:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3.1}$$

$$C_t^f = C_{t-1} * f_t \tag{3.2}$$

Input Gate:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3.3}$$

$$C_t^1 = tanh(W_c[h_{t-1}, x_t] + b_C) \tag{3.4}$$

$$C_t^i = C_t^1 * i_t \tag{3.5}$$

$$C_t = C_t^f + C_t^i \tag{3.6}$$

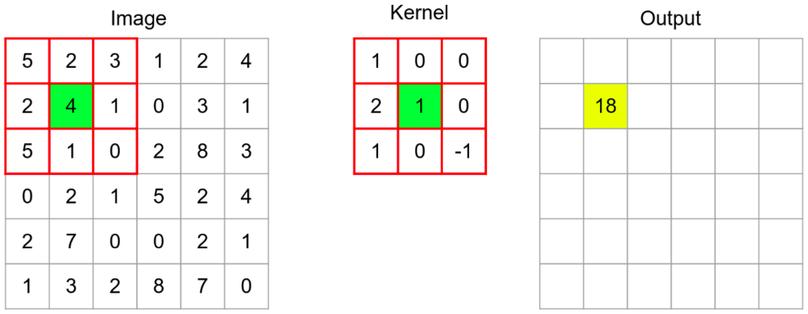
Output Gate:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.7}$$

$$h_t = o_t * tanh(C_t) \tag{3.8}$$

An extension of LSTM is **Bi-directional LSTM**. This network is trained simultaneously in negative as well as positive direction of time. This enables the network to get additional context, from the past as well as the future. Thus networks get familiar with temporal dynamics resulting in improved performance in problems like sequence classification. BILSTM hence are commonly used in tasks where context plays an important role in model performance. Hence, when used in TTS applications, it could infer more contextual speech. [29]

Convolutional Neural Network: CNN first used in image processing showed better results than multilayer perceptron. It uses a lesser number of parameters than that of dense layers. CNN actually emulates how humans emulate their vision. It extracts features such as edges and shapes, scale invariance and translation variance of an image. Basically, CNN is made up of convolution and pooling layers. Convolution is performed between the pixel matrix of images and the kernel. Kernels are a grid of weights; actually they are feature detectors and are learned during training. Hence, convolved weight is given as:



Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	18				

Figure 3.2: Convolution in image by a Kernel

$$\sum_{i=1}^{P} image_i.K_i \tag{3.9}$$

Architectural decision in CNN are impacted by following discussed four parameters:

- **Grid Size:** It is the number of pixels for height and width and are in odd numbers. e.g. 3*3, 5*5.

- **Stride:** Stride is the step size taken while sliding the kernel on the image pixel. It indicates how many pixel kernels are slided on an image pixel.
- **Depth:** Depth refers to the channel of the image. If the image is grayscale then its depth is one i.e. black and white. If the image is RGB then its depth is three.
- **Number of Kernel:** Convolutional layers can have multiple kernels. The output of a single kernel is a single two dimensional array. So, the total number of output is dependent on the total number of kernels.

The second part of CNN is pooling. Pooling layer downsample the image by overlaying a grid on the image. There are two classes of pooling: first is max pooling (i.e. pools maximum value) and second is average pooling (i.e. pools average value.).

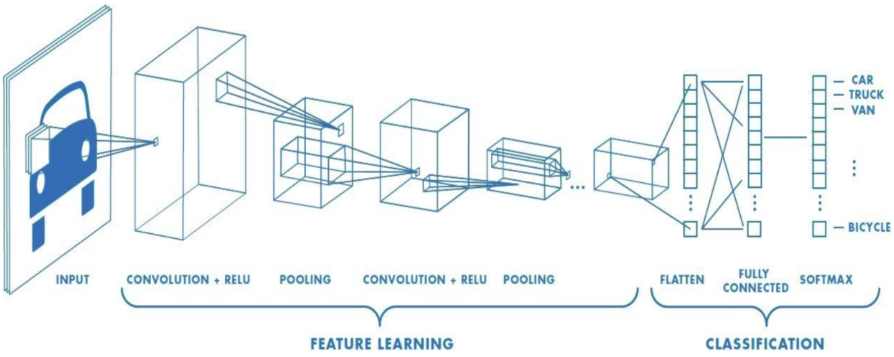


Figure 3.3: Architectural diagram of Convolutional Neural Network [30].

Application in Speech Domain: Basically, the properties of audio signals can be represented using an image by means of Spectrogram or MFCC. This consists of Time information in the x-axis, frequency information in the y-axis and the intensity of pixel value provides the information of Amplitude. These representations can be used in tasks such as music genre classification, WaveNet vocoder and so on.

3.2 Background from Digital Signal Processing

Vibrating objects causes air molecules to oscillate as a result it changes air pressure, hence sound is produced. Sound being analog mechanical waves, we want to process it via digital device. So, Sampling and Quantization are two steps followed by Analog to Digital converter to convert these analog signals into digital format. There are three levels of audio features one needs to know before diving into the Audio AI.

- **High Level:** Instrumentation, Melody, Key, Rhythm, Genre.
- **Mid Level:** MFCCS, fluctuation patterns, pitch, beat.
- **Low Level:** Amplitude, frequency, spectrum (i.e. time frequency representation).

Spectral Leakage: As the signal is being converted from the time domain to frequency domains, endpoints of the signal are discontinuous. Hence, the processed signal is not an integer number of periods. This results in spectral leakage. This problem can be addressed by techniques called **Windowing**. In order to eliminate the samples at both ends of the frame, a windowing function is applied to each frame. Thus, it generates a periodic signal. Most popular window used in DSP is Hann Window. For K_{th} sample, windowing function is given as:

$$w(k) = 0.5.(1 - cos(\frac{2\pi k}{K-1})), k = 1....K \tag{3.10}$$

$$s_w(k) = s(k).w(k), k = 1...K \tag{3.11}$$

But this creates another problem i.e. while joining the frame after windowing, we lose the information at the start and end of the frame. This problem can be solved by using overlapping frames. So, we come to a new term called **Hop Length**. Hop length decides the amount of shift in the right direction when tackling the next frame. For overlapping conditions, the hop length is less than the frame length and this difference in length gives us the overlap length.

Mel-spectrogram: Basically, a group of single frequency audio waves comprise audio signals. The mathematical tool known as Fourier transform transforms mathematical representation of signal into the frequencies that make that signal and also represent frequency amplitude. This conversion of time domain signal into frequency domain signal is known as spectrum. To obtain the spectrogram, Fourier transform is computed on segments of signal i.e. window known as short time Fourier transform. Such obtained representation is known as spectrogram. But the following three features are supposed to be represented by the spectrograms, but the spectrogram could not represent all features. Hence, Mel-scale spectrograms are developed.

- Time Frequency Representation.
- Amplitude Representation in perceptually relevant manner.
- Frequency Representation in perceptually relevant manner.

Mel-scale tries to mimic the human way of perception of sound. Humans have better tendencies to tell difference in lower frequencies than that of higher. In the mel spectrogram, the frequencies in the spectrogram are converted into the mel scale. Following are steps to obtain a Mel spectrogram.

- Extract short time Fourier transform
- Convert amplitude to decibels
- Convert frequencies into Mel scale

MFCCs: Mel-Frequency Cepstral Coefficients consist of two distinct words, Mel-Frequency and Cepstral. Cepstral comes from the words Spectrum and provides the information about how spectral bands change. So, in this analysis four terms gets introduced i.e. Cepstrum, Quefrency, Liftering and Rhamonic. The term cepstrum was introduced in the 1960s during the study of echoes in seismic signals and from the 2000s, it is used for the task of music information retrieval.

Cepstrum can be computed using:

$$C(x(t)) = F^{-1}[log(F[x(t)])] \tag{3.12}$$

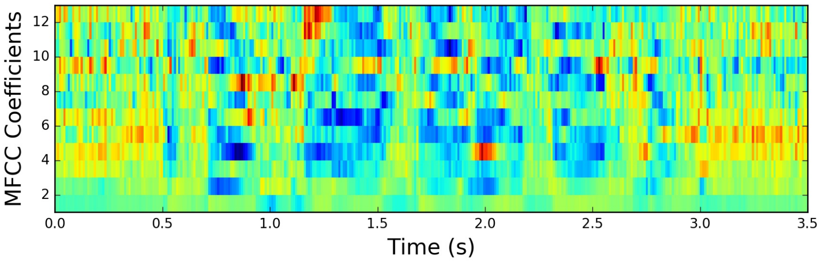


Figure 3.4: Mel-frequency Cepstral Coefficients

Where, $x(t)$: Time domain signal
$F[x(t)]$ gives spectrum of time domain signal.

Also the frequency in Hertz can be converted into mel frequency using:

$$Mel(f) = 2595log(1 + \frac{f}{700}) \tag{3.13}$$

So, **Cepstrum** is the spectrum of spectrum. Basically, humans produce sound by passing a glottal pulse via the vocal tract. Vocal tract condition these glottal pulses in its frequency response in order to generate speech. Hence, speech is convolution of vocal tract frequency response with glottal pulse. Cepstrum helps us to separate these components.

3.3 Performance Improvement Methods in Neural Networks

Regularization and Batch Normalization are two common methods for improving the performance of neural networks. There are two techniques used for regularization:

- **Dropout:** During the training of a network of neurons, certain neurons either in visible units or in hidden units are dropped out such that reduced network is left during training in certain epochs. This mechanism is termed as dropout. The probability of keeping neurons is p and dropping a certain neuron is $1 - p$. Edge to and from those neurons are removed. This techniques solves the issues of overfitting in neural network and is one of the most popular method for regularization. [31]

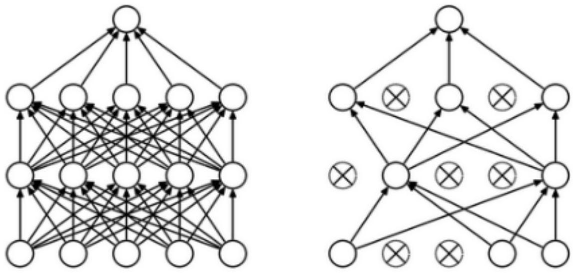


Figure 3.5: Dropout: a. Standard Neural Network b. Dropout applied on Neural Network [31].

- **Zoneout:** In contrast to zero masking techniques in Dropout these techniques use identity masks. During different cycles of timestep, this technique stochastically forces certain portions of hidden units to maintain their last state values. Similar to that of dropout, zoneout also makes use of random noise for training a pseudo-ensemble for improving generalization. But by saving instead of dropping the hidden units of network, state information and gradient information are more willingly propagated through the time, as in feed forward stochastic depth networks. [32]

Batch Normalization: Batch Normalization decreases the training time and improves the model accuracy. It is generally used to overcome the problem of internal covariate shift, hence used to normalize the input data of each layer. Before training, we perform preprocessing on input data such that we normalize the data such that it is normally distributed. So, two main reason i.e. for data preprocessing which helps to avoid the early saturation in non linear activation function such as sigmoid activation function and secondly data preprocessing assure that the input data being fed in the network are in same range of values and this helps to improve model accuracy as well as reduce computational complexity. But the values in the intermediate layers (i.e. hidden layers) are continuously changing. This is due to constantly changing distribution in the activation function. This reduces the speed of training as every layer needs to learn to adapt to the newly created distribution in each training epoch. Hence, batch normalization needs to be done before activation layer. [33]

3.4 Evaluation Methods for Synthesized Audio

For the result validation, we will be following both the qualitative and quantitative approach. In a qualitative approach, mean opinion score is the most trusted approach, hence will be used. It's a very difficult problem to quantitatively rate the quality of audio. No single method can score the similarity of the audio. We will be scoring the rendered audio sample by zero crossing method as well as by measuring the correlation of the generated audio sample with the audio sample in the test set.

- **Mean Opinion Score:** It is an average of scores given by subjects to represent the quality of the system or quality of experience. It measures how natural the system generated voice is.

- **Zero Crossing:** Zero crossing is popularly used for measuring noise in the signal and can be extended to measure how different the synthesized audio is from the test set. It is mostly used in systems such as music information retrieval and speech recognition. It is the number of times any signal crosses its horizontal axis and can be obtained using following equation:

$$ZCR_t = \frac{1}{2} . \sum_{k=t.K}^{(t+1).K-1} |sng(s(k)) - sgn(s(k+1))| \tag{3.14}$$

- **Correlation:** In signal processing, cross-correlation is a measure of similarity of two series as a function of the displacement of one relative to the other. The cross-correlation is similar in nature to the convolution of two functions. In an autocorrelation, which is the cross-correlation of a signal with itself, there will always be a peak at a lag of zero, and its size will be the signal energy.

CHAPTER 4

METHODOLOGY

4.1 Block Diagram

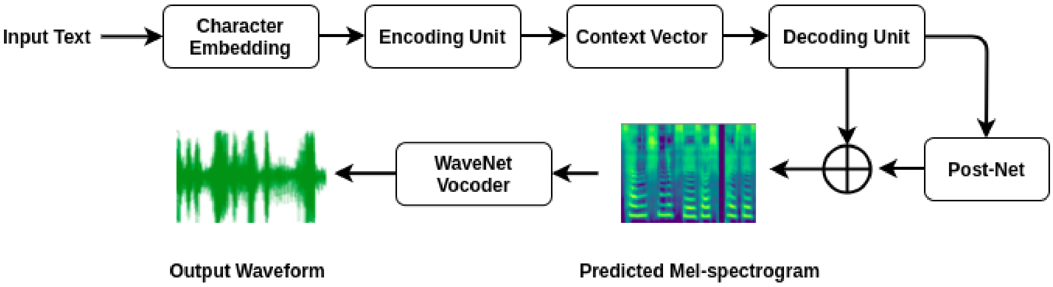


Figure 4.1: An overall diagram for end-to-end TTS synthesis using feature prediction network followed by Convolutional Neural Vocoder i.e. WaveNet.

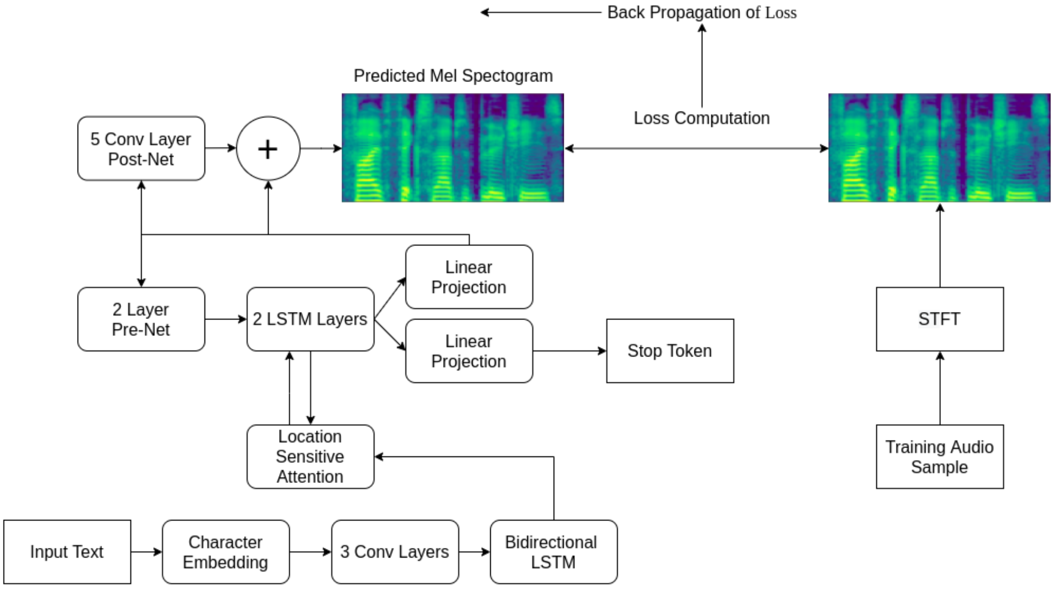


Figure 4.2: A detailed internal block diagram for training a model to predict mel-spectrogram representation network from input text [9].

4.2 Data Collection and Preprocessing

Building a good model for TTS requires a large dataset of high quality audio. This is why TTS for only a few languages are developed. This research uses a dataset from OpenSLR. OpenSLR is a site devoted to hosting speech and language resources, such as training corpora for speech recognition, and software related to speech recognition. OpenSLR consists of Nepali speech and language resources which can be used to experiment our setup. The dataset is collected by recording in a minimal noise environment (i.e. in recording studio), hence contains high quality transcribed audio in wav-format. Audio samples were collected in 2016, 2017 and 2018 by google team in Nepal [11] [12]. Here transcribed noise free audio is preserved and the noisy and untranscribed audio are removed. Also the data is arranged on speaker wise folder structure with respective script stored in csv format. The text data needs to be normalized before training. Also, the number of audio dataset per speaker is less in number for training deep neural networks, hence speaker adaptation is not applicable. Character embedding unit uses 512 dimensions for input text representation.

SN	Details	No of Hours	No of speaker	No of dataset	Resource Code
1	Large Nepali ASR training dataset [12].	2.47	19	2,064	SLR43
2	High Quality TTS dataset for Nepali [11].	165	527	157,000	SLR54

Table 4.1: Characteristics of Collected Sounds.

For the low level acoustic representation, a mel-scale spectrogram is used because of two properties: First, its representation is smoother than the waveforms and secondly, it is easier representation for training as the spectrogram is phase invariant within each frame. Since the spectrogram discards the phase information, WaveNet instead of the traditional Griffin-lim algorithm, can be implemented.

Training process involves two steps, the first one is the feature prediction network which is followed by the output waveform inference network. A low-level acoustic representation is chosen for this work: mel frequency spectrograms, to bridge the two components. Using a representation that is easily computed from time-domain waveforms allows to train the two components separately. This representation is also smoother than waveform samples and is easier to train using a squared error loss because it is invariant to phase within each frame.Hence, first we need clean recorded transcribed audio for successful implementation of our proposed method. The first stage of training involves building a model such that for given input, it predicts the mel-spectrogram. This obtained spectrogram is normalized by compressing its dynamic range. Here the goal is to use deep neural net for mapping character token into the spectrogram representation of audio version for provided text. Tokenization of text converts Nepali text into a sequence of numeric values [34]. A mel-frequency spectrogram is related to the linear-frequency spectrogram, i.e. the short-time Fourier transform (STFT) magnitude. It is obtained by applying a non linear transform to the frequency axis of the STFT, inspired by measured responses from the human auditory system, and summarizes the frequency content with fewer dimensions. Using such an auditory frequency scale has the effect of emphasizing details in lower frequencies, which are critical to speech intelligibility, while de-emphasizing high frequency details, which are dominated by fricatives and other noise bursts and generally do not need to be modeled with high fidelity.These spectrogram are also represented in numeric array. So, these two sequences of numbers are obtained in numpy arrays i.e. .npy files. Hence, the model is trained to predic the spectrogram array from text token. But there might arise the issue of dimension mismatch. This can be addressed by padding the short sequences.

4.3 Encoder-Decoder Architecture

First part of Architecture is about the feature prediction network. It is a seq-to-seq model along with attention. Seq-to-seq architecture is a neural network where input and output units are in sequence. The intuition behind the encoder-decoder architecture is to read the input sequential data such as sentence and then compress that specific portion of sentence into the fixed length thought vector. The decoder is trained in such a way that the input thought vector is decoded back one step at a given time. Original encoder-decoder uses RNN for this task.

This fixed length representation is a bottleneck for encoder decoder architecture. Hence, an attention mechanism was introduced to address the problem of summarizing the input into fixed length by encoder. In attention mechanisms, given input sequences of text are mapped into a series of annotations. These annotations are the same length as that of the input sequence. The decoder learns to attend different subsets of these annotations for output generation.

4.3.1 Encoder

Consider input sequence $X = (X_1, X_2, X_3, ..., X_T)$. Then the encoder maps these sequences to a series of annotations of same length, say $H = (H_1, H_2, H_3, ..., H_T)$. So this hidden state at H_i has the information of respective input token X_i.

Here each **Character** is represented as X_i i.e. Input Token. **Encoder** block consists of a block of **Three Convolutional Layers** which is followed by **Bidirectional LSTM layer.** Convolution in image represents local correlations among input features. In this scenario, convolution finds local correlations between input characters by extracting feature maps. This is useful for RNN modeling of input sequences. Hence, long term context can be maintained in the convolution. Say two words Flour and Floor, both of these words have similar starting words but pronounced differently. Humans pronouncing these words take into consideration the character u. This can be captured by RNN but it's hard to maintain such long term dependencies in a practical scenario. Also the use of convolution increases the effectiveness of the model by making it robust towards silent characters such as in case of ”django”, character ”d” is silent and so on. It's better to capture N-grams by the use of convolutional networks before feeding character embedding into the RNN layer.

Mathematically,

$$X = (X_1, X_2, ..., X_{T_x}), X_i \epsilon R^{K_x} \tag{4.1}$$

$$H = (H_1, H_2, ..., H_{T_x}), H_i \epsilon R^{K_x} \tag{4.2}$$

where K_x represents vocabulary size of input tokens and T_x represents the length of input sentence. The embedded character of the input sentence is first convolved using three convolution filters, say F_1, F_2 and F_3. ReLU non linearity is used at each layer. Say E be embeddings.

$$f_{e,i} = Relu(F_3 * Relu(F_2 * Relu(F_1 * EX))) \tag{4.3}$$

Passing these features via bi-directional LSTM generates the hidden state of the encoder.

$$H = EncoderRecurrency(f_e) \tag{4.4}$$

Bidirectional RNN is used to infer information of input characters from both the past and the future. Hence H_j summarizes both past and future words. Additionally, RNN tends to remember present context more hence H_j would be more focused on X_j. The final encoder output is the concatenated result of forward as well as backward hidden states. This character embedding layer is followed by a stack of 3 convolutional layers. Each layer have 512 filters having shape of 5*1. These filters span 5 characters whose output is conditioned to Batch Normalization followed by ReLU Activations. The output of these layers is passed via Bi-directional LSTM having 256 units in each direction. Bi-directional LSTM is supposed to produce encoded features.

4.3.2 Attention Mechanism

Attention mechanism is the direct link between decoder output and encoder outputs. Attention mechanism is meant to pay attention to the output of the encoder, such that the decoder gets most relevant information for present state output. From a programming point of view, attention network is a multilayer feed forward neural network which learns two things, first it learns to predict one output at a time and second it learns to align the output to its respective inputs. So, there comes the end of a traditional method ineffective for long sequence, compressed fixed length representation of input to thought vector.

Content based attention computes a context vector out of encoder outputs to determine the most important context. For this, energy of each vector is computed and passed through the softmax activation. Based on the output of activation, attention to specific vectors is determined for given context.

Location based attention is similar to content based attention but the only difference is the methods for computing the energy/score. Say e_{ij} be score, then

$$e_{ij} = score(\alpha_{i-1}, h_j) = v_a^T tanh(Wh_j + Uf_{i,j}) \tag{4.5}$$

where $f_{i,j}$ represents location features. It is computed by convolution of previous stage alignment α_{i-1} with convolution filters. Other parameters are weight to be learned. Rather than content of input tokens, location based attention is concerned about the location and distance of tokens.

4.3.3 Decoder

First, the previous spectrogram frame is fed into the **prenet layer**. Context vector that is computed on the previous decoding stage is concatenated with the output of prenet. This concatenated output is fed into the RNN decoder. This step is known as input feeding. Hence, the RNN generates a query vector which is used for the computation of the new context vector. This new context vector is concatenated with the out of decoder and fed into the projection layers. The projection layer predicts the spectrogram frame and also computes stop token probability. Projection layers project the hidden states of RNN to the output space. Stop token probability determines whether to stop decoding or not. Prenets is used to translate frames in frequency space to the hidden state such that RNN can predict next frames and also helps in regularization of networks.

Mathematically,

$$py_i = Prenet(y_{i-1}) = Relu(W_2 Relu(W_1 y_{i-1} + b_1) + b_2) \tag{4.6}$$

$$s_i = LSTM(LSTM(s_{i-1}, py_i, c_{i-1})) \tag{4.7}$$

$$y_i = Linear([s_i, c_i]) = W_p[s_i, c_i] + b_p \tag{4.8}$$

$$y_{s,i} = Sigm(Linear([s_i, c_i])) = \sigma(W_s[s_i, c_i] + b_s) \tag{4.9}$$

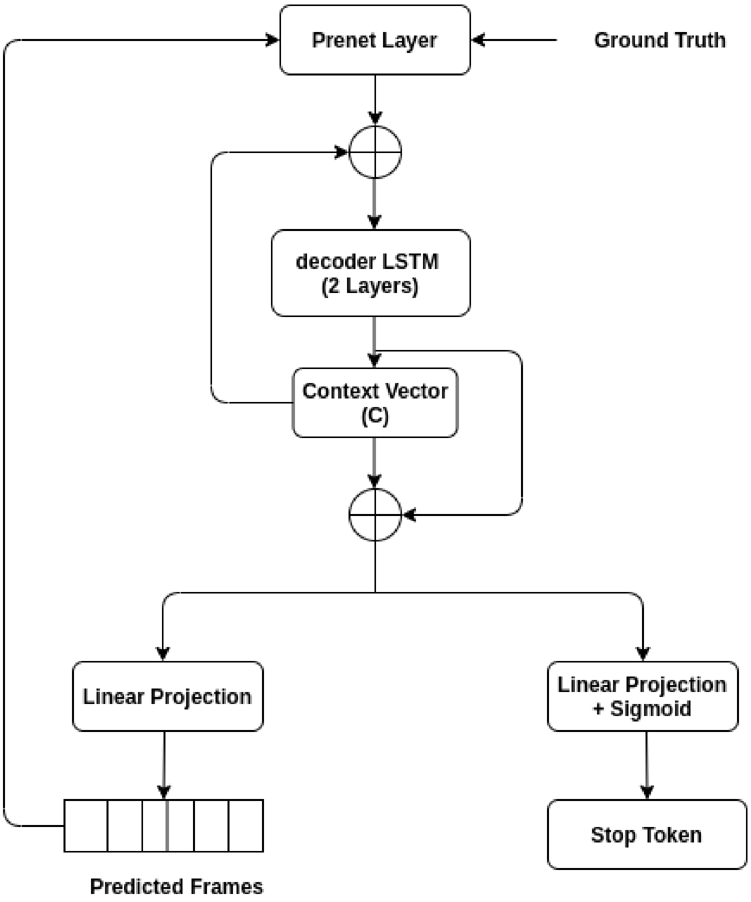


Figure 4.3: An Architectural diagram for Decoder.

The fully connected layers of prenet have 256 hidden ReLU units. The attention context vector and the output of Pre-Net is propagated through LSTM layers. This LSTM layer has two layers which are unidirectional with 1024 units each.

Post-Net: After the decoding stage, the synthesized Mel spectrogram is fed into the Post-Net. Post-Net is a stack of 5 convolution layers with non-linearities. Since, the convolution layer can capture the features from both past and future context in sequence, the network learns from past data. Hence, it improves the performance of the network. Sum of decoder spectrogram and residual gives final output. Each post-net layer is made up of 512 filters of shape 5*1 with batch normalization.

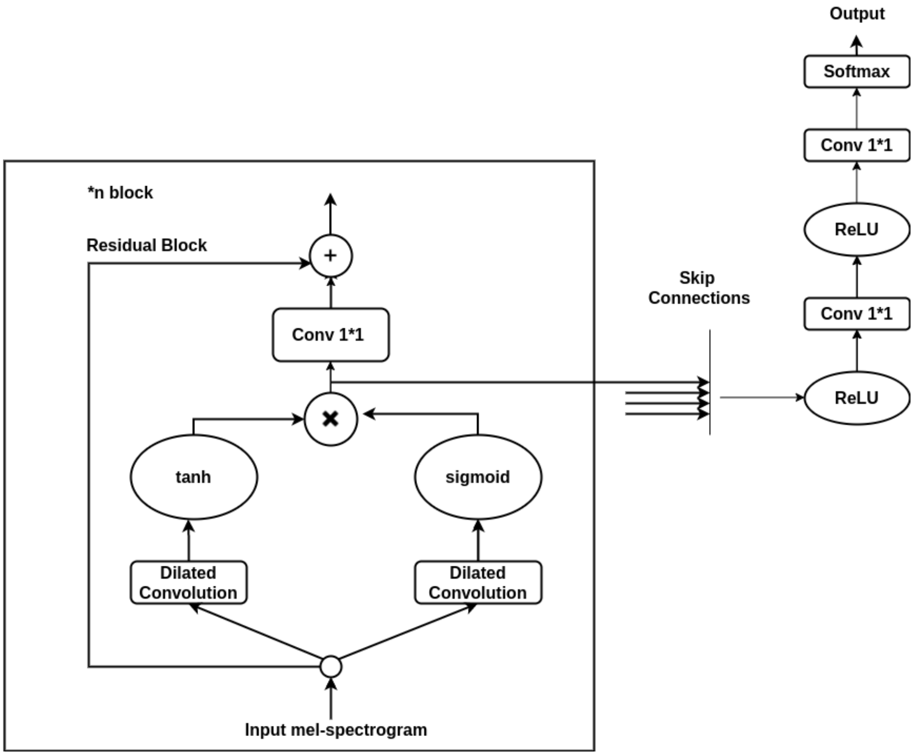


Figure 4.4: A detailed internal block diagram of WaveNet vocoder for training a model to predict waveform samples from input mel-spectrogram [1]

4.3.4 WaveNet Vocoder

WaveNet is used for inverting the spectrogram representation into audio signals of time-domain waveforms. Inverse STFT can also be used for the purpose but WaveNet is supposed to maintain prosody of speech signal and syntheisze more natural audio. Neural vocoders are neural networks that generate (speech) waveforms from acoustic feature inputs. WaveNet has been suggested for Text-To-Speech synthesis showing that a non-linear autoregressive system can mimic speech generation very well, if it is appropriately locally conditioned with linguistic information. WaveNet is an autoregressive network, which generates a probability distribution of the next sample given some segment of previous samples. The next sample is produced by sampling from this distribution. An entire sequence of samples is produced by feeding previously generated samples back into the model. A defining feature of WaveNet is that it uses of dilated convolution in order to increase the receptive field exponentially with the number of layers. It explores raw audio generation techniques, inspired by recent advances in neural autoregressive generative models that model complex distributions such as images and text. WaveNet architecture is able to model the distributions over thousands of random variable.

During training of WaveNet as vocoder, Mel spectrogram is used as input as it emphasizes on low frequency details which is crucial for clarity of speech. Additionally, the speech samples are dynamic range compressed via µ-law transformation and then quantized to 8-bits.The above architecture works on the following steps.

- **Dilated Convolution Layer :** It referred to the convolution with holes. It consist of convolution with a large filter where zeros are filled to increase the receptive field of input. It is similar to that of convolution with a large filter which is derived from the original filter by dilating it with zeros, but is significantly more efficient. A dilated convolution allows the network to function on a coarser scale than with a normal convolution. It is similar to that of pooling or strided convolutions, but the difference is the output has the same size as the input. The dilated convolution with dilation 1 yields the standard convolution.
- **Gated Activation Units :** Dilated convolution is followed by tanh and sigmoid function. This forms expert (i.e. Dilated Convolution + tanh nonlinearity) and the gate (Dilated Convolution + sigmoid nonlinearity). These blocks are combined via element wise multiplication.
- **Residual Block and Skip Connections :** The purpose of these two blocks is to speed the output convergence and also enable deeper training of models.
- **Softmax Distribution :** Softmax distribution has a more flexible categorical distribution, hence can model the arbitrary distribution. It is to be noted that mu-law companding transformation is used to make the sequence more tractable. Hence, the output will be 256 possible instead of 65,536.

Hence, the end-to-end network for synthesis of speech from text is shown below:

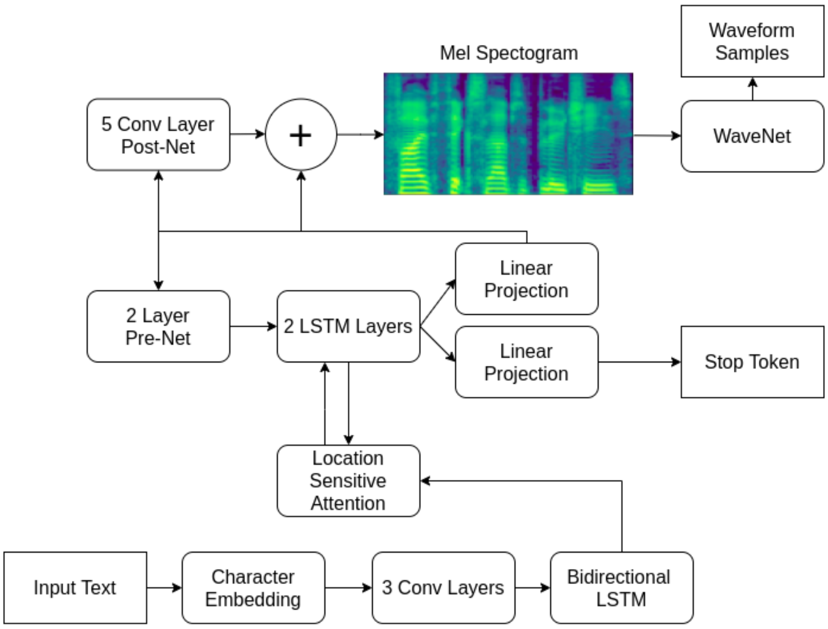


Figure 4.5: An Overall Architectural diagram for Text-to-Speech synthesis [9]

4.3.5 Evaluation Methodology :

For evaluation of built model, the aim is to evaluated the result by MOS (i.e. qualitative evaluation) and Correlation Measure (i.e. quantitative measures).

Mean Opinion Score : In MOS, the score has five levels of ratings from 1 to 5 for labels of bad, poor, fair, good and excellent. Audio samples in a test set are rendered from the model from input text. Then the original output and rendered output are rated by eight independent human listeners. Finally the averaged output of ground truth and rendered sample is evaluated. Say N is the number of subjects and R is their respective ratings. Then:

$$MOS = \frac{\sum_{n=1}^{N} R_n}{N} \tag{4.10}$$

Correlation : Cross-correlation is generally used when measuring information between two different time series. It ranges from -1.0 to +1.0. The closer the cross-correlation value is to 1, the more closely the sets are identical. Correlation of the two signal can be computed using:

$$SignalValidation = correlationof(abs(fft(x)), abs(fft(x_1))) \tag{4.11}$$

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Training Setup

The training procedure involves the training of the feature/spectrogram prediction network which is followed by a neural WaveNet vocoder for predicting waveform using input features from the output of the first network. The training setup for the Text-to-Speech model was done in AWS. Deep Learning AMI (Ubuntu 18.04) version 46.0 was used. The p3.2x large instance was chosen which has 8 core virtual cpu with 61 GB memory. The instance consists of pre-installed TensorFlow 2.4, CUDA and runs on the NVIDIA V100 GPU. Both network training was performed on a 157,000 dataset of male speakers using a batch size of 32 for seventy thousand epochs. Dataset was approximately 165 hours long. Spectrogram prediction network was trained for 6.5 days and WaveNet was trained for another 4 days. Both stages inherit some common hyper-parameters and are as below:

Common Hyper-parameters

Sample Rate = 22050

Window Size = 1100

Hop size = 256

Frame Shift in ms = 12.5

Non-uniform Fast Fourier Transform = 2048

Number of frequency = 1025

5.2 Experiment 1: Feature Prediction Network

5.2.1 Model Building

The components of the feature prediction network were laid as per the architectural diagram in methodology section. Embedding hidden size of 512 dimension is taken followed by same dimensional convolution filters . The kernel size of 5*5 was used for the encoder. ReLU activation serve as activation function in encoder section of the network. The model was trained for 70,000 epochs. Following are the hyperparameters used in this feature prediction network.

Learning Rate = 0.0001
Optimizer = Adam
Batch Size = 32
Embedding Dropout Probability = 0.1
Encoder Convolution Filters = 512
Encoder Convolution Kernel Size = 5
Encoder LSTM units = 256
Number of Prenet Layers = 2
Prenet Units = 256
Prenet Activation = ReLU
Decoder LSTM Units = 1024
Attention Dimension = 128
Attention Kernel = 31
Postnet Convolution Filters = 512
Postnet Convolution Kernel Sizes = 5

5.2.2 Evaluation of the Model

The data used in training was collected from professional sound recording studio. This class of data was preprocessed to subdue noise inherited in the signal. Also the data volume was good enough for training the neural network. Large data size of good quality and higher number of training epochs are two contributing factor to substantially reduce the training loss of the model. We can see that attention loss was almost zero for both training and testing sets. This attest to the proper implementation of sequence-to-sequence model. Also, the spectrogram prediction network has very low loss. During training it was nearly zero but evaluation on unseen data showed the loss of 0.22. So the model appears to be stable and suitable candidate for speech synthesis task.

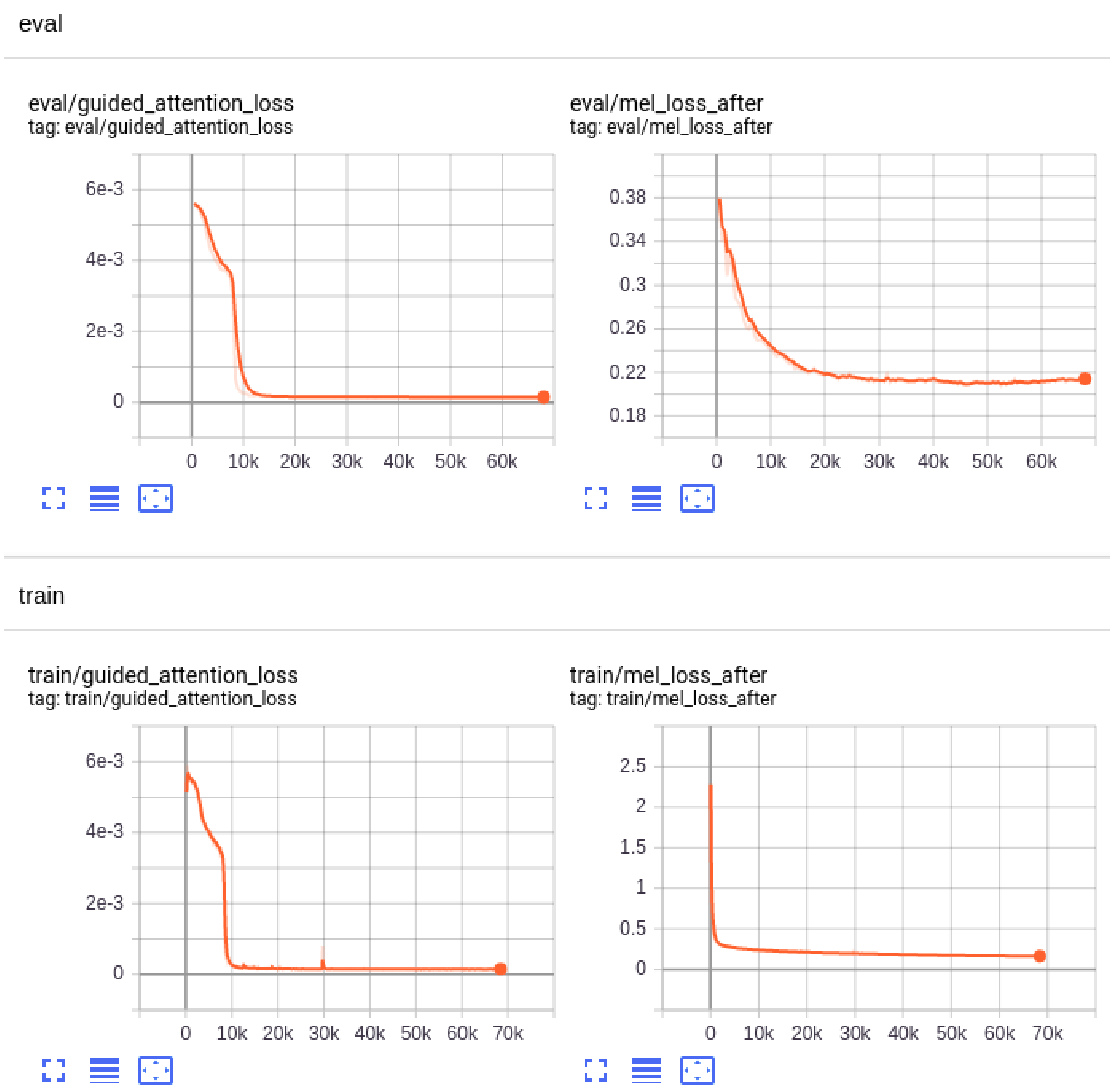


Figure 5.1: Loss vs Number of Epochs, Plot for Attention Component and Feature Prediction Model Loss.

5.3 Experiment 2: Waveform Synthesis Network

5.3.1 Model Building

Mel-spectrogram representation demolish the phase information. So, the major task of this block is to recover those lost information such that the natural prosody of the speech is recovered during inference. This model is trained on the ground truth-aligned predictions from the feature prediction network. When generating speech in inference mode, the ground truth targets are not known. Therefore, the predicted outputs from the previous step are fed in during decoding, in contrast to the teacher-forcing configuration used for training. This network is trained on 157,000 dataset for 80,000 training steps. Following hyperparameters were used for training the waveform synthesis network.

Learning rate = 0.00001

Batch Size = 8

Optimizer = Adam

Dropout = 0.05

Number of Dilated Convolution = 24

Number of Dilated Convolution stacks = 4

Number of Residual Block = 256

Kernel Size = 3

5.3.2 Evaluation of the Model

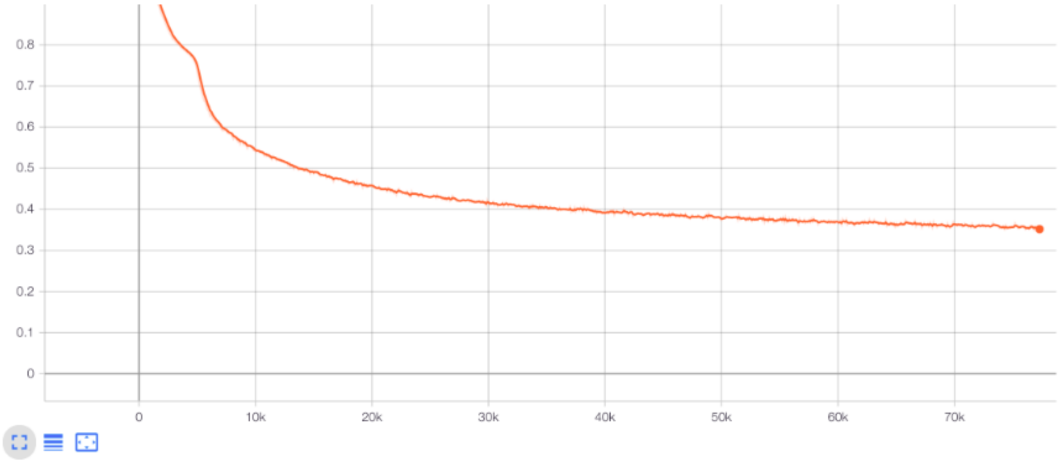


Figure 5.2: Loss vs Number of Epochs, Plot for Training Loss of WaveNet Vocoder.

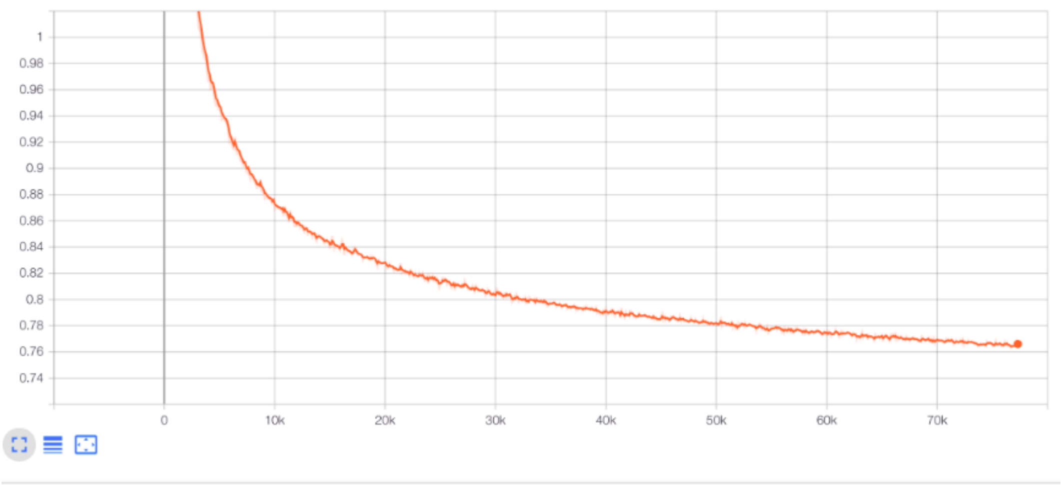


Figure 5.3: Loss vs Number of Epochs,Plot for Validation Loss of WaveNet Vocoder.

The training loss for the model is 0.38 and validation loss is 0.763. Loss below one is considered as good indicator for the WaveNet Vocoder. Data scientist from google trained WaveNet for 500,000 steps to obtain near human level performance. But such training is computationally expensive and not viable to train with the currently available resource. We can obtain acceptable model quality by training on fewer number of steps.

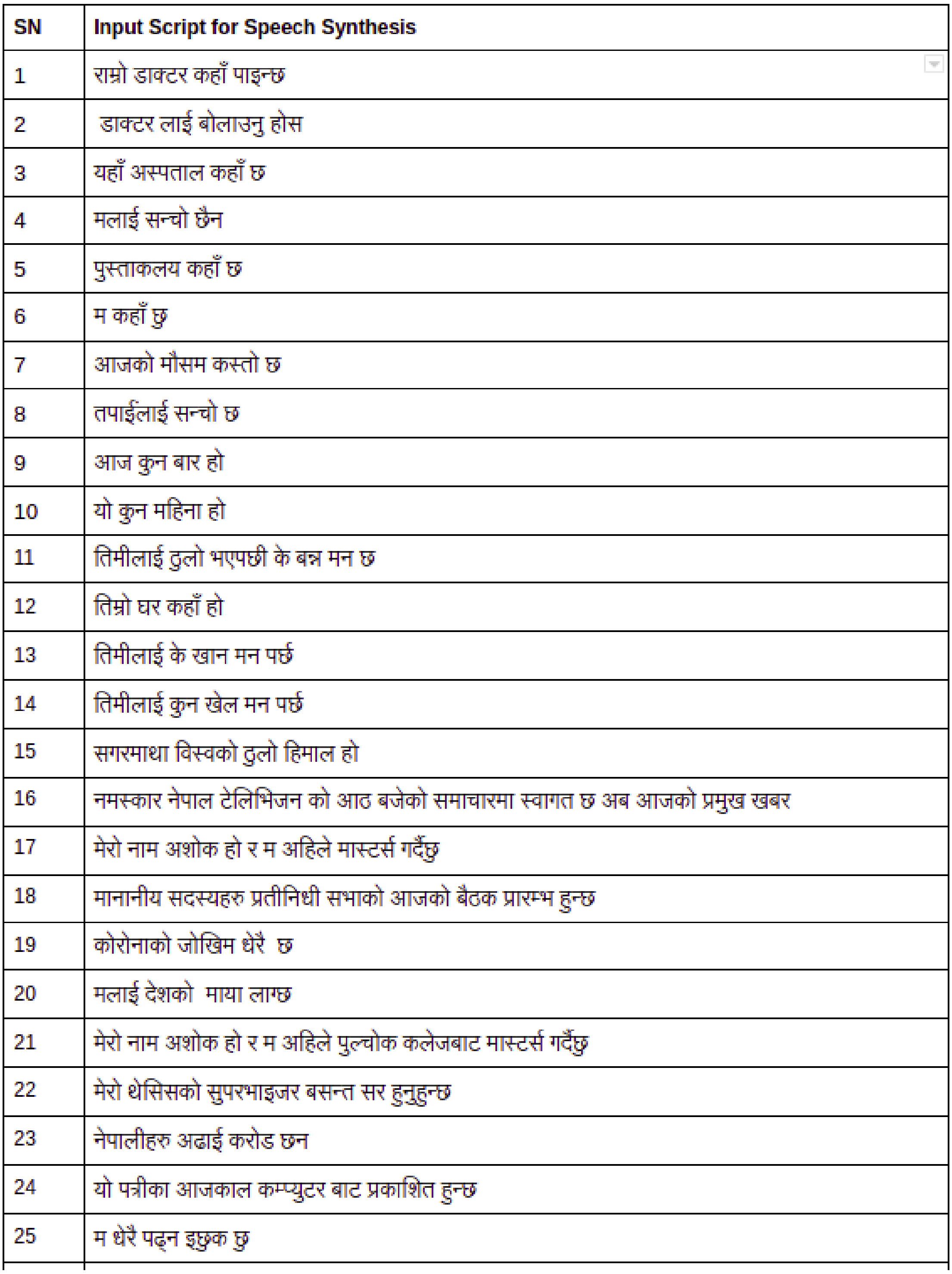
5.4 Synthesis of Speech from Text

The feature prediction model was trained for seventy thousand epochs. The sound synthesis from the model was done by the input of plain Nepali Unicode text. As the model was not trained on numerals, dates, symbols, currency and so on, it synthesized ambiguous speech in such cases. In such a scenario, it is necessary to perform text cleaning.

- **Model can Synthesize :** Plain text which is input in Nepali unicode format, numbers that are converted into text (i.e. spelled out).
- **Model cannot Synthesize:** Number, Symbols, Currency, Date and Time, telephone numbers, years, percentage, emails, computer programs, urls. If these data are cleaned and spelled out, then these can be read by the model.

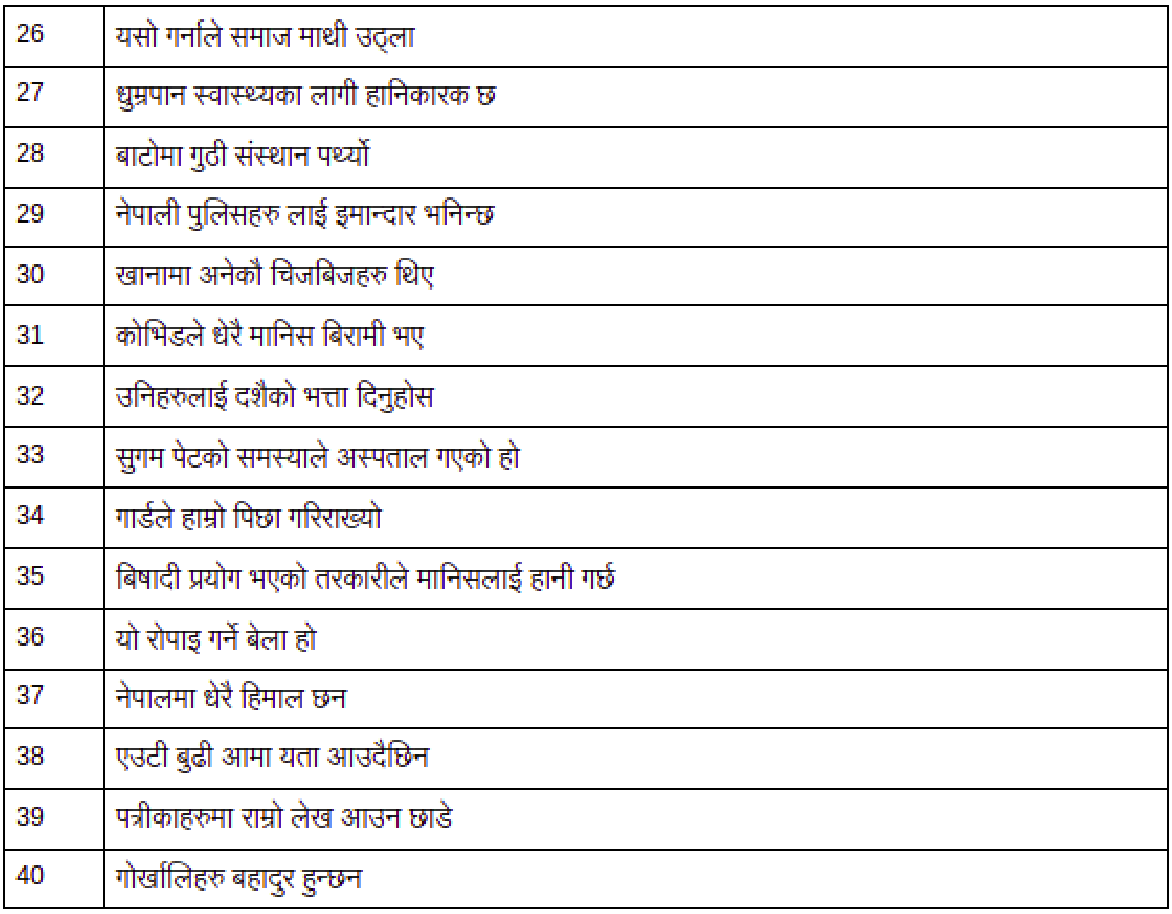
Following sets of script were synthesized from the model and kept on the **google**

drive [1]. Despite these scripts, there are some complex words to pronounce. So, in the later section we have also compared synthesized words with the recorded spoken word.



SN	Input Script for Speech Synthesis
1	राम्रो डाक्टर कहाँ पाइन्छ
2	डाक्टर लाई बोलाउनु होस
3	यहाँ अस्पताल कहाँ छ
4	मलाई सन्चो छैन
5	पुस्ताकलय कहाँ छ
6	म कहाँ छु
7	आजको मौसम कस्तो छ
8	तपाईलाई सन्चो छ
9	आज कुन बार हो
10	यो कुन महिना हो
11	तिमीलाई ठुलो भएपछी के बन्न मन छ
12	तिम्रो घर कहाँ हो
13	तिमीलाई के खान मन पर्छ
14	तिमीलाई कुन खेल मन पर्छ
15	सगरमाथा विस्वको ठुलो हिमाल हो
16	नमस्कार नेपाल टेलिभिजन को आठ बजेको समाचारमा स्वागत छ अब आजको प्रमुख खबर
17	मेरो नाम अशोक हो र म अहिले मास्टर्स गर्दैछु
18	मानानीय सदस्यहरु प्रतीनिधी सभाको आजको बैठक प्रारम्भ हुन्छ
19	कोरोनाको जोखिम धेरै छ
20	मलाई देशको माया लाग्छ
21	मेरो नाम अशोक हो र म अहिले पुल्चोक कलेजबाट मास्टर्स गर्दैछु
22	मेरो थेसिसको सुपरभाइजर बसन्त सर हुनुहुन्छ
23	नेपालीहरु अढाई करोड छन
24	यो पत्रीका आजकाल कम्प्युटर बाट प्रकाशित हुन्छ
25	म धेरै पढ्न इछुक छु

[1] Drive link for synthesized audio from text :
https://drive.google.com/drive/folders/1efljV2GkmqvZ53yozRCXRf-EcnobcRcy?usp=sharing



26	यसो गर्नाले समाज माथी उठ्ला
27	धुम्रपान स्वास्थ्यका लागी हानिकारक छ
28	बाटोमा गुठी संस्थान पर्थ्यो
29	नेपाली पुलिसहरु लाई इमान्दार भनिन्छ
30	खानामा अनेकौं चिजबिजहरु थिए
31	कोभिडले धेरै मानिस बिरामी भए
32	उनिहरुलाई दशैंको भत्ता दिनुहोस
33	सुगम पेटको समस्याले अस्पताल गएको हो
34	गार्डले हाम्रो पिछा गरिराख्यो
35	बिषादी प्रयोग भएको तरकारीले मानिसलाई हानी गर्छ
36	यो रोपाइ गर्ने बेला हो
37	नेपालमा धेरै हिमाल छन
38	एउटी बुढी आमा यता आउदैछिन
39	पत्रीकाहरुमा राम्रो लेख आउन छाडे
40	गोर्खालिहरु बहादुर हुन्छन

Table 5.1: Script for Synthesizing Audio from the Model.

5.5 Overview on Synthesized Speech

From the above script, our text-to-speech model synthesized the audio and we did cross checking on accurateness of the pronounced words by listening to those synthesized. The output audio appears to be noise free but appears to have some pronunciation issues. Character त sounds similar to ट . There are also some issue with the pronouncing of ”12 Khari”. The model sometimes confuses words such as कलेज and is synthesized as कोलेज . This ambiguity might be due to the ambiguity of pronunciations of character for different sentence structure. Instead of feeding the spelled out number, directly number were feed for the test purpose. This sometimes synthesized noise while most of the time the model generated silence frame. But there appears no issue on evaluation of word such as मेरो , पुल्चोक , हुनुहुन्छ and so on.

5.6 Evaluation on Complex Words

5.6.1 Qualitative Evaluation

There are some identified Nepali words which are harder to pronounce. In this section we will look at the waveform and Mel-spectrogram of both synthesized and recorded speech and visually evaluate them.

- **Plot for synthesized word ऋषि .**

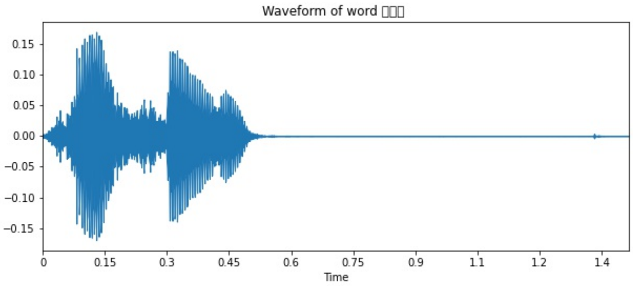


Figure 5.4: Waveform Plot of Synthesized Word ऋषि .

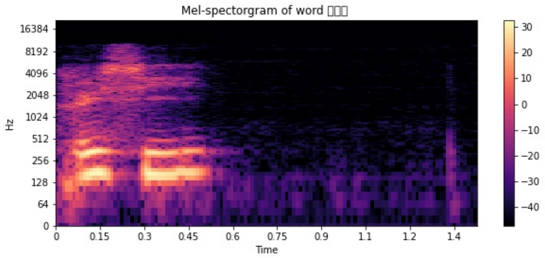


Figure 5.5: Mel-spectrogram Plot of Synthesized Word ऋषि .

- **Plot for recorded word ऋषि .**

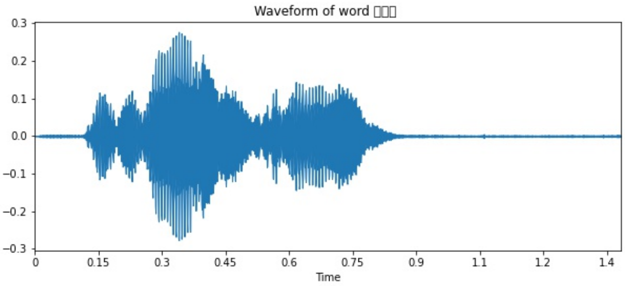


Figure 5.6: Waveform Plot of Recorded Word ऋषि .

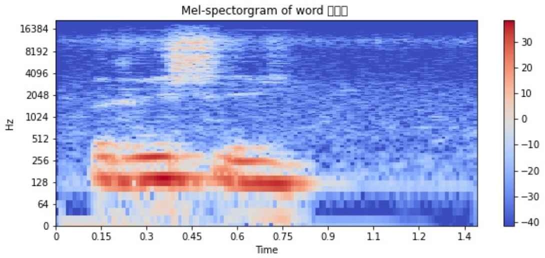


Figure 5.7: Mel-spectrogram Plot of Recorded Word ऋषि .

- **Plot for synthesized word समृधी .**

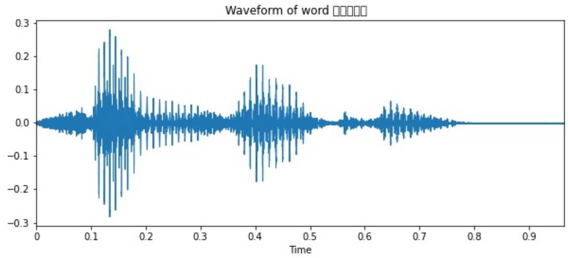


Figure 5.8: Waveform Plot of Synthesized Word समृधी .

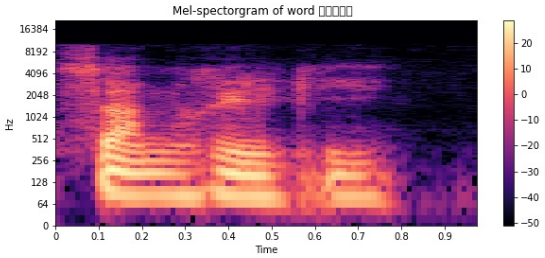


Figure 5.9: Mel-spectrogram Plot of Synthesized Word समृधी .

- **Plot for recorded word समृधी .**

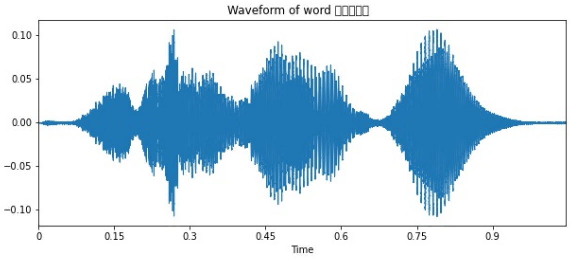


Figure 5.10: Waveform Plot of Recorded Word समृधी .

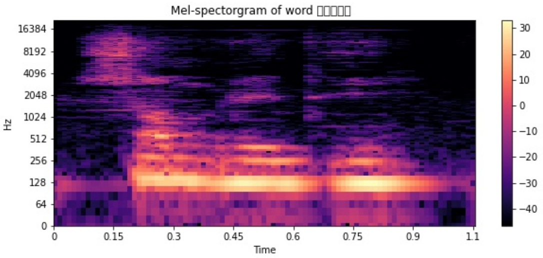


Figure 5.11: Mel-spectrogram Plot of Recorded Word समृधी .

As these two words, other various words can be synthesized and analyzed. Human vocal tract produce sound by passing the different amplitude glottal pulse through the vocal tract. These pulses are conditioned to filters hence certain response forms some specific speech. Words like ऋषि and समृधी have complex responses, hence they are harder to pronounce as well as to synthesize by traditional approach.

Humans listen to audio on a log scale, not via linear scale. Let's say the first pair of audio have 500 and 520 Hz and another pair have 1000 and 1020 HZ. Although both pairs have a frequency difference of 20Hz, we can feel more difference in lower frequency than that of higher frequency. So, we are more concerned about the Mel-spectrum representation. We can see the variations of amplitude in waveform for recorded and synthesized speech and found out that they have closely similar variation patterns. This is more clarified by Mel-spectrogram plot. The level of brightness represents the amplitude and we see that similar pattern in brightness of pixel for both classes of speech. Also, the similarity in formants indicates similar kinds of content and timbre information. In the first sample, we see different colors in two Mel-spectrograms and this is due to the difference in energy level of recorded and synthesized speech.

5.6.2 Quantitative Evaluation

Each individual has a different style of speaking and it is a complicated process to exactly score the similarity of two audio by mathematical process. In our scenario, we have trained the model on the speech of multiple speakers. So, computation of similarity by correlation of signals is not feasible. Also, computation of mean squared error of two different human speakers is not zero, hence it doesn't measure naturalness of audio. For the closest approach, we computed the cross-correlation of synthesized and recorded speech. For this speech the following text was taken.

SN	Compared Sentence	Correlation
1	ऋषि	76.94 %
2	ऋषिहरु ज्ञानी हुन्छन	81.17 %
3	समृधी	81.88 %
4	समृधी देशको लागी आवश्य कुरा हो	74.25 %

Table 5.2: Characteristics of Collected Sounds.

Hence average correlation in the speech is found to be 78.56 percent. This evaluation can be performed on larger samples of speech for more accurate results.

5.6.3 Evaluation By Mean Opinion Score

MOS is most widely used method for evaluating the quality of audio and video [35]. For the evaluation of our model, we synthesized 40 sample speech from above mentioned script. As per standard, it is necessary to take at least 8 listeners [36] and here we are taking 10. Among these, 8 were male and 2 were female volunteers. They were asked to rate each speech on the basis of four parameters: speech clarity, content correctness, ascent Naturalness and timing exactness [36] . On the basis of quality of speech each parameters were rated from 1 to 5. For this, synthesized speech were shared via google drive and also the rating sheet.

The overall average of all the volunteer score reports MOS of the model to be 3.07. The most close speech synthesis result obtained in English language has reported the MOS of 4.21 and for Chinese 4.08 [1]. Also, the work on sanskrit TTS by using feature prediction network and griffin-lim algorithm shows MOS of 3.18 [37]. Hindi TTS reports MOS 3.9 using Tacotron 2 and waveglow and Our works seems to have good start point for future works and improvement.

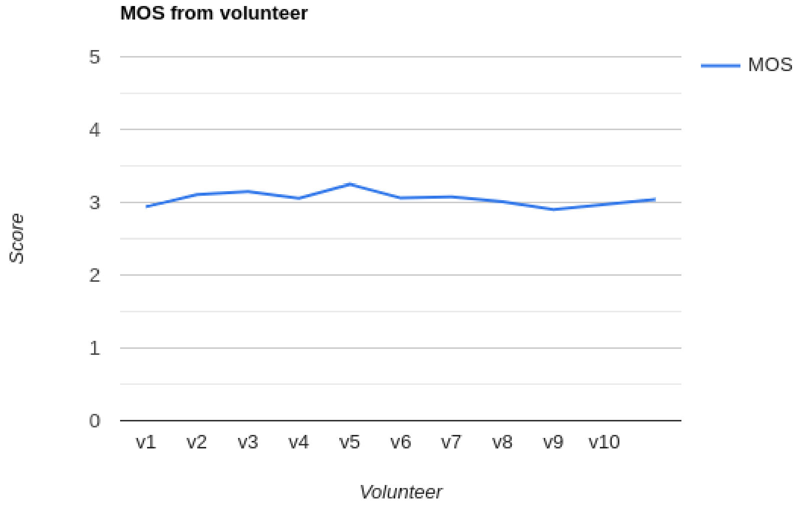


Figure 5.12: Line Chart showing the pattern of score from ten volunteer (v1 to v10.

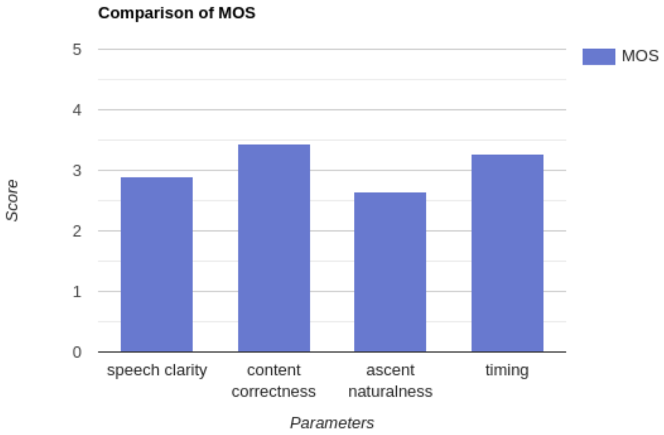


Figure 5.13: Comparison of MOS for parameters contributing on overall score.

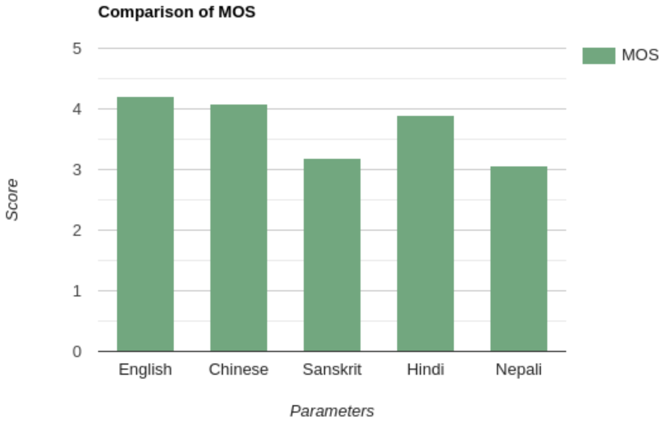


Figure 5.14: Comparison of MOS for different language.

5.7 Utilized Tools and Resources

Text Editor : Visual Studio,

Programming Language : Python 3.7

Framework : Tensorflow, Keras, WaveNet Vocoder

Audio Processing Tools : Librosa 0.8, Audacity

Data Visualization Tools : matplotlib, tqdm (progress tracking), TensorBoard

Server : Amazon Web Service (AWS)

OS : Deep Learning AMI (Ubuntu 18.04) Version 46.0

Support : TensorFlow-2.2, NVIDIA CUDA, cuDNN.

GPU Specification : Nvidia V100.

CPU : 8 core, Memory : 61 GB

CHAPTER 6

EPILOGUE

6.1 Conclusion

The work successfully shown the idea for building end-to-end Nepali Text-To-Speech synthesis network using deep neural networks. The two stage network i.e. feature prediction network followed by WaveNet Vocder, produces audio waveform on input text. The work established the set of hyperparameters which can further be improved by experimentation. The model achieved the MOS of 3.07. Higher accuracy can be achieved by training the network with single speaker dataset. The system is trained directly from data without relying on complex feature engineering and shown the better result than previously developed Nepali TTS [2] system.

6.2 Limitation

Following are the major limitations of this work:

- The model appears to synthesize audio with poor quality of prosody. Due to the voice sample from different cluster of people, the synthesized voice samples have uncomfortable ascent.

- Inferencing of audio from model have higher computational complexity as well as higher time complexity.

- The model cannot synthesize non-standard words such as numerals, symbols and so on. In order to handle such words, it is necessary to develop a Nepali standard text conversion library.

- This system cannot deal with the context of the input text. Let's say we have two context for 1945 i.e. as a number and as a date, in the input text. Then the system is unable to handle such a scenario.

6.3 Future Enhancement

The work must be continued to overcome the shortcomings of this work and can be upgraded to synthesize more natural voice.

- WaveNet Vocoder is still expensive to train. It requires large volume of clean audio sample and powerful GPUs for training. Some work suggests use of WaveGlow [38] can overcome these issues.
- Model can be trained longer and in Multi-GPU. Also, there are many untweaked hyperparameters which can be tuned and experimented for different values. As a result, prosody of speech might improve.
- Transformer Network [39] appears to be highly successful work in neural machine translation. Research can also be directed towards implementing Transformer Network for TTS synthesis.
- Context based Nepali Text-To-Speech system can be developed, which can produce more natural sounding speech.

REFERENCES

[1] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[2] Kaushal Subedi. Nepali text-to-speech, 2015.

[3] Roop Shree Ratna Bajracharya, Santosh Regmi, Bal Krishna Bal, and Balaram Prasain. Building a natural sounding text-to-speech system for the nepali language: research and development challenges and solutions. *Gipan*, 4:106–116, 2019.

[4] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

[5] Sercan Ö Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep voice: Real-time neural text-to-speech. In *International Conference on Machine Learning*, pages 195–204. PMLR, 2017.

[6] Andrew Gibiansky, Sercan Arik, Gregory Diamos, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech. *Advances in neural information processing systems*, 30, 2017.

[7] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. *arXiv preprint arXiv:1710.07654*, 2017.

[8] Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani. Voiceloop: Voice fitting and synthesis via a phonological loop. *arXiv preprint arXiv:1707.06588*, 2017.

[9] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[10] Rupak Raj Ghimire and Bal Krishna Bal. Enhancing the quality of nepali text-to-speech systems. In *Conference on Creativity in Intelligent Technologies and Data Science*, pages 187–197. Springer, 2017.

[11] Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha. Crowd-Sourced Speech Corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali. In *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, pages 52–55, Gurugram, India, August 2018.

[12] Keshan Sodimana, Knot Pipatsrisawat, Linne Ha, Martin Jansche, Oddur Kjartansson, Pasindu De Silva, and Supheakmungkol Sarin. A Step-by-Step Process for Building TTS Voices Using Open Source Data and Framework for Bangla, Javanese, Khmer, Nepali, Sinhala, and Sundanese. In *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, pages 66–70, Gurugram, India, August 2018.

[13] Oliver Watts, Cassia Valentini-Botinhao, Felipe Espic, and Simon King. Exemplar-based speech waveform generation. In *INTERSPEECH*, pages 2022–2026, 2018.

[14] Simon King. An introduction to statistical parametric speech synthesis. *Sadhana*, 36(5):837–852, 2011.

[15] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[16] Manish K Ssarma, Avaas Gajurel, Anup Pokhrel, and Basanta Joshi. Hmm based isolated word nepali speech recognition. In *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 1, pages 71–76. IEEE, 2017.

[17] Bhusan Chettri and Krishna Bikram Shah. Nepali text to speech synthesis system using esnola method of concatenation. *International Journal of Computer Applications*, 62(2), 2013.

[18] Pratistha Malla. *Nepali Text to Speech using Time Domain Pitch Synchronous Overlap Add Method*. PhD thesis, IOE, 2015.

[19] Nepali-TTS: Full manual of Nepali. tts. Springer, 2008.

[20] Richard Sproat, Alan W Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. Normalization of non-standard words. *Computer speech & language*, 15(3):287–333, 2001.

[21] Daniel Watson, Nasser Zalmout, and Nizar Habash. Utilizing character and word embeddings for text normalization with sequence-to-sequence models. *arXiv preprint arXiv:1809.01534*, 2018.

[22] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[23] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[25] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243, 1984.

[26] Akira Tamamori, Tomoki Hayashi, Kazuhiro Kobayashi, Kazuya Takeda, and Tomoki Toda. Speaker-dependent wavenet vocoder. In *Interspeech*, volume 2017, pages 1118–1122, 2017.

[27] Corentin Tallec and Yann Ollivier. Unbiasing truncated backpropagation through time. *arXiv preprint arXiv:1705.08209*, 2017.

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[29] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[30] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee, 2017.

[31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[32] David Krueger, Tegan Maharaj, János Kramár, Mohammad Pezeshki, Nicolas Ballas, Nan Rosemary Ke, Anirudh Goyal, Yoshua Bengio, Aaron Courville, and Chris Pal. Zoneout: Regularizing rnns by randomly preserving hidden activations. *arXiv preprint arXiv:1606.01305*, 2016.

[33] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[34] Bharat Bhatta, Basanta Joshi, and Ram Krishna Maharjhan. Nepali speech recognition using cnn, gru and ctc. In *Proceedings of the 32nd Conference on Computational Linguistics and Speech Processing ({ROCLING} 2020)*, pages 238–246, 2020.

[35] Robert C Streijl, Stefan Winkler, and David S Hands. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227, 2016.

[36] Mahesh Viswanathan and Madhubalan Viswanathan. Measuring speech quality for text-to-speech systems: development and assessment of a modified mean opinion score (mos) scale. *Computer speech & language*, 19(1):55–83, 2005.

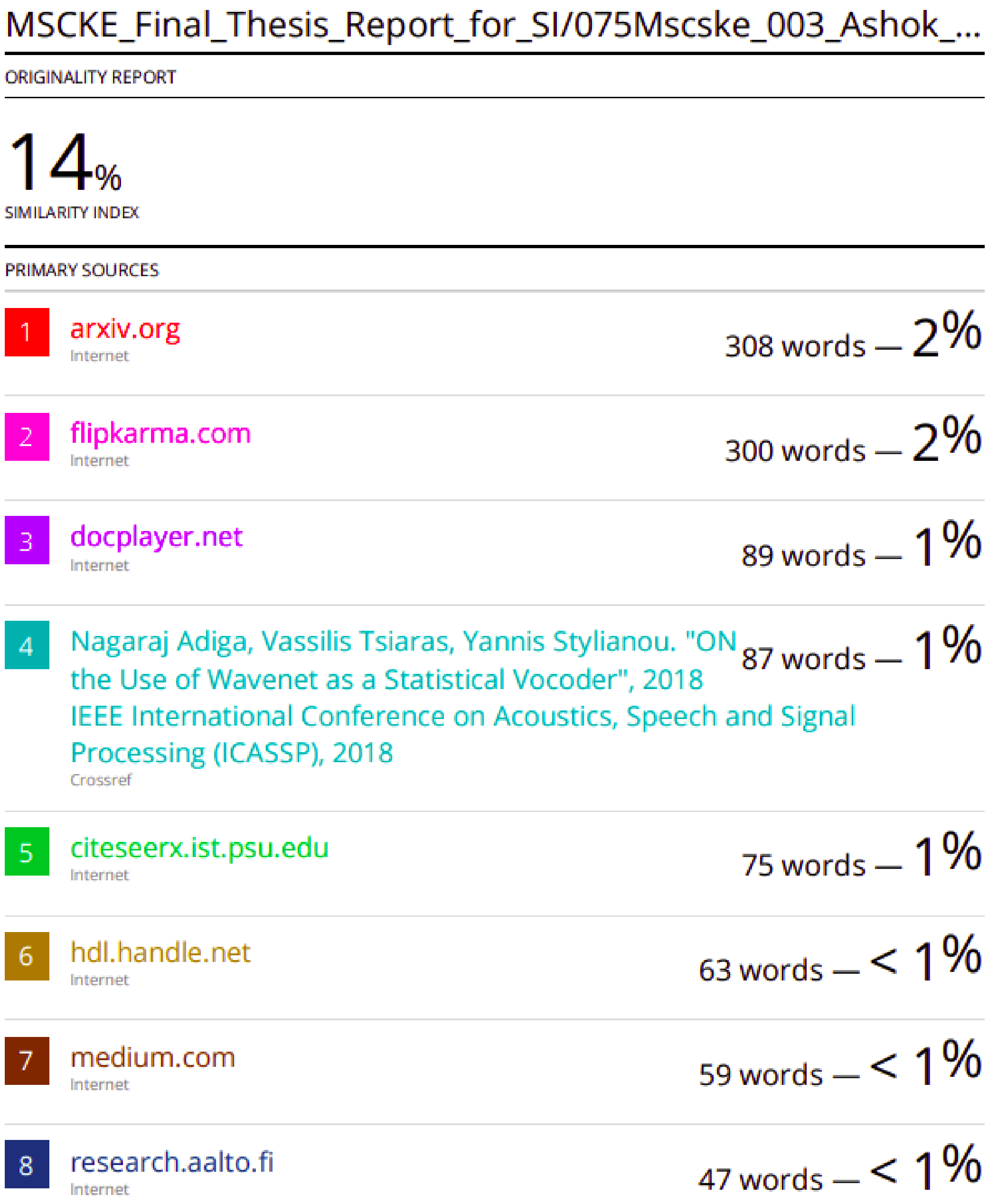
[37] Ankur Debnath, Shridevi S Patil, Gangotri Nadiger, and Ramakrishnan Angarai Ganesan. Low-resource end-to-end sanskrit tts using tacotron2, waveglow and transfer learning. In *2020 IEEE 17th India Council International Conference (INDICON)*, pages 1–5. IEEE, 2020.

[38] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[39] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6706–6713, 2019.

APPENDIX A

Similarity Report



MSCKE_Final_Thesis_Report_for_SI/075Mscske_003_Ashok_...

ORIGINALITY REPORT

14%

SIMILARITY INDEX

PRIMARY SOURCES

1	arxiv.org Internet	308 words — 2%
2	flipkarma.com Internet	300 words — 2%
3	docplayer.net Internet	89 words — 1%
4	Nagaraj Adiga, Vassilis Tsiaras, Yannis Stylianou. "ON the Use of Wavenet as a Statistical Vocoder", 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018 Crossref	87 words — 1%
5	citeseerx.ist.psu.edu Internet	75 words — 1%
6	hdl.handle.net Internet	63 words — < 1%
7	medium.com Internet	59 words — < 1%
8	research.aalto.fi Internet	47 words — < 1%