Transformer-based Nepali Text-to-Speech

Ishan Dongol and **Bal Krishna Bal**
Information and Language Processing Research Lab
Kathmandu University, Dhulikhel, Nepal
ishandongol@gmail.com
bal@ku.edu.np

Abstract

Research on Deep learning-based Text-to-Speech (TTS) systems has gained increasing popularity in low-resource languages as this approach is not only computationally robust but also has the capability to produce state-of-the-art results. However, these approaches are yet to be significantly explored for the Nepali language, primarily because of the lack of adequate size datasets and secondarily because of the relatively sophisticated computing resources they demand. This paper explores the FastPitch acoustic model with HiFi-GAN vocoder for the Nepali language. We trained the acoustic model with two datasets, OpenSLR and a dataset prepared jointly by the Information and Language Processing Research Lab (ILPRL) and the Nepal Association of the Blind (NAB), to be further referred to as the ILPRLNAB dataset. We achieved a Mean Opinion Score (MOS) of 3.70 and 3.40 respectively for the same model with different datasets. The synthesized speech produced by the model was found to be quite natural and of good quality.

1 Introduction

Deep learning-based TTS has achieved significant popularity and success recently. One of the reasons for the increasing popularity is the limited need for manual feature engineering compared to traditional methods like formant, concatenative, and statistical parametric speech synthesis (Kumar et al., 2023). The quality of the TTS systems in resourceful languages like English, Chinese, Hindi, etc. seems to be comparatively much better, where the efforts seem to be focused on making the models computationally efficient as well as producing natural-sounding speech. Unfortunately, Nepali TTS research is still in its early stages, and even the existing results require further improvements. Some of the limited Nepali TTS researches include (Bajracharya et al., 2018; Basnet, 2021; Basnet et al., 2021, 2023).

As far as deep learning-based TTS research is concerned, most of the research is conducted on autoregressive models like WaveNet (van den Oord et al., 2016), Tacotron 2 (Shen et al., 2018). However, these models are computationally expensive and require large datasets. (Basnet, 2021; Khadka et al., 2023). Non-autoregressive models, on the other hand, are able to synthesize Mel spectrograms in the order of magnitude faster than autoregressive ones. The reason behind this is that in non-autoregressive models, the RNNs-based acoustic model has been replaced by a Transformer-based one. This apparent advantage of the non-autoregressive model demonstrated by FastSpeech (Ren et al., 2019) and FastPitch (Łańcucki, 2021) and the results achieved by (Kumar et al., 2023) encouraged us also to explore the model for Nepali language as well.

In this paper, we report our attempt to train the FastPitch model for the Nepali language with the two different datasets available to us. We also point to different areas of improvement on the trained model in the future.

The paper comprises 7 sections which are structured in the following order: Section 1 provides the background of the research. Section 2 discusses about the existing TTS systems. Section 3 explains the datasets, models, and frameworks used. Section 4 explains about the evaluation process we adopted. Section 5 consists discussion about the findings and comparison between existing Nepali TTS systems. Section 6 provides information on improving the developed TTS system.

2 Related Works

One of the recent Nepali TTS (Bajracharya et al., 2018) used concatenative speech synthesis and unit selection process to generate a natural-sounding voice. They successfully built a Nepali TTS that was also used along with the Non-Visual Desktop

Access (NVDA). However, still a few issues of overlaps and echoes were reported in the generated speech, which was suspected to be caused by the potential mislabelings between recorded speech phones and the corresponding transcribed characters. (Bajracharya et al., 2018).

More recent research has been conducted with autoregressive models like WaveNet (van den Oord et al., 2016) and Tacotron 2 (Shen et al., 2018) (Basnet, 2021; Basnet et al., 2021; Khadka et al., 2023; Basnet et al., 2023).

(Basnet, 2021) trained WaveNet on SLR43 (Sodimana et al., 2018) and SLR54 (Kjartansson et al., 2018). The model achieved a good result but the generated audio lacks the quality of prosody due to the voice sample from different clusters of people. The output has an uncomfortable accent (Basnet, 2021). Similarly, (Basnet et al., 2021) trained WaveNet on custom dataset. However, the results are poor due to the noisy data and insufficient training epoch.

Following the trends, (Basnet et al., 2023) and (Khadka et al., 2023) trained Tacotron 2 with WaveGlow (Prenger et al., 2018) and HiFi-GAN (Kong et al., 2020) respectively. (Khadka et al., 2023) achieved the highest MOS score but the model only generates 1-14 seconds of audio (Khadka et al., 2023) As per the authors, the naturalness of the generated audio can further be improved with more high-quality training data.

Similarly, (Kumar et al., 2023) rigorously explored various TTS systems for 13 Indian languages across choices of acoustic models, vocoders, supplementary loss functions, training schedules, and speaker and language variations. Out of which FastPitch (Łańcucki, 2021) and HiFi-GAN (Kong et al., 2020) performed best (Kumar et al., 2023).

3 Research Methodology

In this section, we explain the datasets that we used in training the model. Besides, we also provide a detailed overview of a sequence of processes followed and the frameworks adopted as part of developing the FastPitch model for Nepali. As shown in Figure 1, the Research Methodology can be described by four major stages, viz., Literature Review, Dataset Preparation, Training, and Evaluation. The detailed overview of dataset preparation, and training strategy is shown in Figure 2, and Figure 3 respectively.

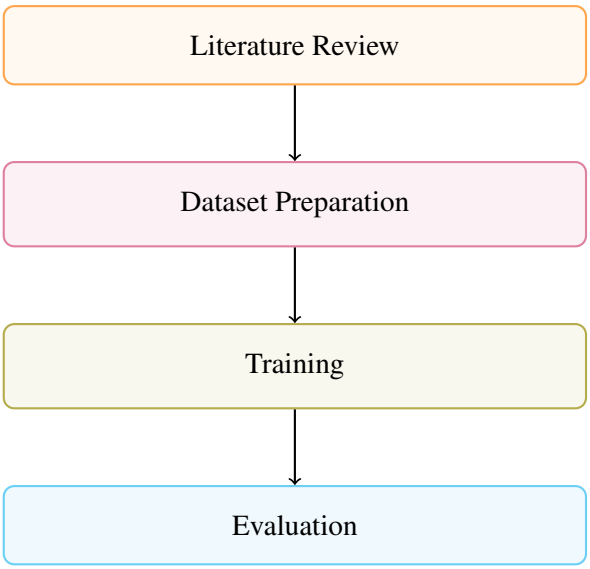

Figure 1: Overview of Research Methodology

3.1 Datasets

In this research, we experiment with two different datasets as shown in Table 1. To differentiate the output of the FastPitch model on two different datasets, we further coined the models, Danphe and Munal, which are essentially the model versions trained on the ILPRLNAB and later fine-tuned with the SLR43 datasets respectively.

Dataset	Utterances	Duration (hours)
ILPRLNAB	4460	5.07
SLR43	2064	2.80

Table 1: Total utterance and duration of dataset used.

3.1.1 ILPRLNAB

For training the Danphe model, we use the dataset prepared jointly by ILPRL and NAB. We use 4460 normalized utterances for our experiment. The audio was recorded at 16.00 kHz on a mono channel in a quiet studio setting with the voice of a professional Nepali speaker (Bajracharya et al., 2018). All the audio samples were upsampled to 22.05 kHz. We also enhanced the audio quality for the last 500 epochs with the help of dolby.io [1]. The statistics of the dataset are shown in Table 2.

3.1.2 SLR43

For the Munal model we use the High-quality TTS data for Nepali (SLR43) (Sodimana et al., 2018) prepared by Google for the Nepali language. The

[1] https://dolby.io/

Figure 2: Dataset Preparation before training

dataset consists of 2064 multi-speaker utterances. The audio was recorded at 48.00 kHz. All the audio samples were downsampled to 22.05 kHz. The statistics of the dataset are shown in Table 3. We show the dataset preparation process in Figure 2.

3.2 Text Processing

We create a text formatter using Coqui-TTS[2] that simply generates the dataset configuration for both multi-speaker and single-speaker dataset. As the transcripts were pre-processed separately, we omitted the text cleaning process prior to the training.

3.3 Training & Inference

We use the open-source Coqui-TTS library to implement our model. We use the default learning rate scheduling of the library. However, we use different batch sizes for training the Danphe and the Munal models. We use batch sizes of 32 and 16 for Danphe and Munal models respectively. The Danphe model is trained on ILPRLNAB (See Section 3.1.1) dataset for 2500 epochs. The Munal model is a finetuned version of the Danphe model with the training conducted for 2500 epochs using the SLR43 (See Section 3.1.2) dataset. All the models are trained on NVIDIA GeForce RTX 3080 Ti 12 GB GPU. We train FastPitch with the Adam optimizer along with the following parameters - $\beta_1 = 0.99$ and $\beta_2 = 0.998$ with weight decay of $\lambda = 10^{-6}$ and learning rate $\alpha = 10^{-4}$.

[2]https://github.com/coqui-ai/TTS

Statistics	Duration (seconds)
Mean	4.09
Standard Deviation	1.15
Minimum	1.18
25^{th} Percentile	3.24
Median	3.99
75^{th} Percentile	4.86
Maximum	10.97

Table 2: Statistics of audio clips of the ILPRLNAB dataset.

Statistics	Duration (seconds)
Mean	4.88
Standard Deviation	2.15
Minimum	1.40
25^{th} Percentile	3.27
Median	4.39
75^{th} Percentile	6.10
Maximum	13.84

Table 3: Statistics of the audio clips of SLR43 dataset.

IndicXlit for Transliteration: We use the IndicXlit (Madhani et al., 2022) transliteration model to convert Unicode text to Romanized text. IndicXlit is a single transformer-based multilingual transliteration model for Roman to Indic script conversion. It supports 21 Indic languages, achieves state-of-the-art results on the Dakshina (Roark et al., 2020) test set, and establishes strong baselines on the Aksharantar test set (Madhani et al., 2022).

FastPitch for Acoustic Modeling: FastPitch is a fully parallel text-to-speech model based on FastSpeech (Ren et al., 2019), conditioned on fundamental frequency contours. Its architecture is based on FastSpeech, composed mainly of two feed-forward Transformer (FFT) stacks (Łańcucki, 2021). It is the best-performing model for Indic languages among other models experimented in

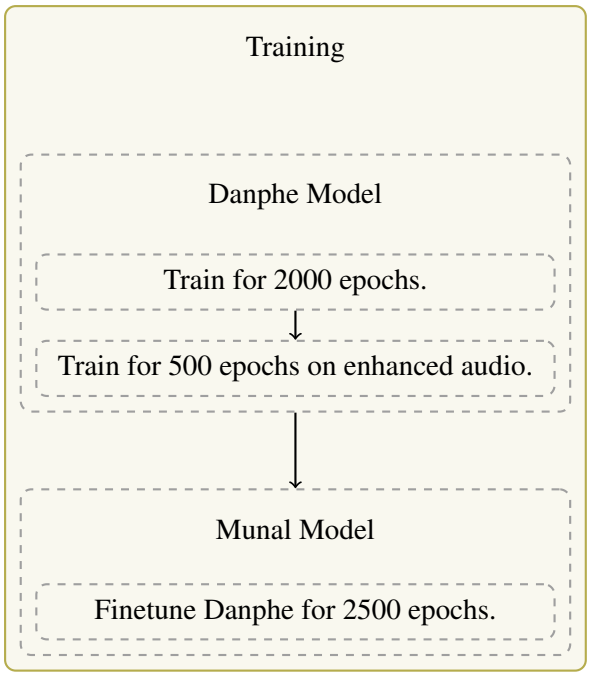

Figure 3: Overview of Training

(Kumar et al., 2023). In addition, it is robust to alignments (Łańcucki, 2021) which is ideal for low-resource language like Nepali. The architecture is shown in Figure 4.

FastPitch is able to rapidly synthesize high-fidelity Mel-scale spectrograms with a high degree of control over the prosody and leads to state-of-the-art results without any overhead. (Łańcucki, 2021).

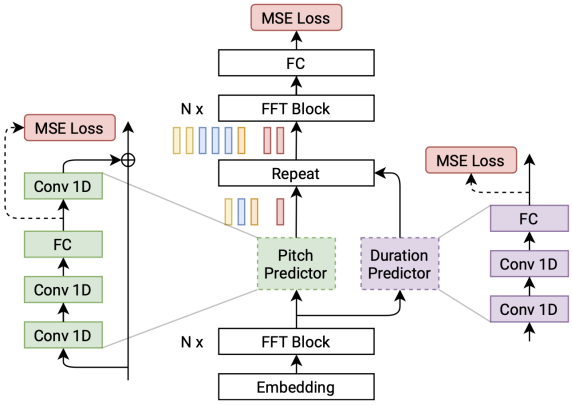

Figure 4: Architecture of FastPitch (Łańcucki, 2021) similar to FastSpeech (Ren et al., 2019).

Alignment via Alignment Learning Framework: For text-to-audio alignment, we use the Alignment Learning Framework (Badlani et al., 2021) which is an extension to the alignment learning approach proposed in RAD-TTS (Shih et al., 2021). The framework enables non-autoregressive models to be trained without relying on external aligners.

The framework improves the alignment convergence speed of existing attention-based mechanisms, simplifies the training pipeline, and makes the models more robust to errors on long utterances. Most importantly, the framework improves the perceived speech synthesis quality, as judged by human evaluators (Badlani et al., 2021).

HiFi-GAN Vocoder: In the final stage, to convert the mel spectrogram to waveform, we use HiFi-GAN vocoder model (Kong et al., 2020). It outperforms autoregressive and flow-based models both in terms of efficiency and accuracy. A subjective human evaluation (MOS) of a single speaker dataset indicates that HiFi-GAN demonstrates similarity to human quality while generating 22.05 kHz high-fidelity audio 167.9 times faster than real-time on a single V100 GPU (Kong et al., 2020).

Due to the lack of high computing resources we used the pre-trained HiFi-GAN (Kong et al., 2020) vocoder model trained on the LJ Speech Dataset (Ito and Johnson, 2017) English dataset.

4 Evaluation

We evaluate our models using subjective MOS on a validation dataset of 20 utterances unseen during training. MOS is a widely-used metric in which human listeners rate the quality of voice samples on a scale, typically from 1 (worst) to 5 (best), providing an average score that reflects the perceived quality of synthesized speech. Out of 24 volunteers, only 13 of the volunteers rated all the audio. As a result, only the responses of 13 volunteers were used to calculate the MOS. The result of the evaluation along with the existing TTS models is shown in Table 4.

5 Results and Discussion

We were able to achieve MOS scores of 3.70 and 3.40 respectively for the Munal and Danphe models. Danphe was trained using 5.07 hours of data and Munal was trained using an additional 2.80 hours of data resulting in a total of 7.87 hours. The generated output samples can be accessed via `https://ishandongol.com.np/transformer-tts/`

It took us approximately 1.5 days to train Danphe's acoustic model for 2500 epochs with a batch size of 32 on our GeForce RTX 3080 Ti 12 GB GPU. As per (Basnet, 2021) it took them 6.5 days to train the acoustic model and an additional 4 days

Models	MOS		Architecture	Dataset
AB	3.07		WaveNet	SLR43 (2018) & SLR54 ((2018))
ABE	2.79		WaveNet	Own
A	3.60		Tacotron 2 & WaveGlow	SLR43 (2018)
S	4.04*		Tacotron 2 & HiFi-GAN	SLR43 (2018) & Own
D **(Ours)**	3.40 ± 0.95		FastPitch & HiFi-GAN	ILPRLNAB (2018)
	Speaker 1	**Speaker 2**		
M **(Ours)**	3.70 ± 0.92	3.51 ± 0.91	FastPitch + HiFi-GAN	SLR43 (2018)

Table 4: MOS, Architecture and Dataset of Models Ashok Basnet (AB) (Basnet, 2021), Ashok Basnet, et. al. (ABE) (Basnet et al., 2021), Aawaj (A) (Basnet et al., 2023), Shruti (S) (* Post processed audio) (Khadka et al., 2023), Danphe (D), and Munal (M).

TTS Models	Training Strategy	Vocoder
AB	Trained from scratch	Trained from scratch
ABE	Trained from scratch	Trained from scratch
A	Fine-tuned English Model	Pre-trained on LJ (2017)
S	Fine-tuned English Model	Pre-trained on LJ (2017)
D **(Ours)**	Trained from scratch	Pre-trained on LJ (2017)
M **(Ours)**	Fine-tuned D	Pre-trained on LJ (2017)

Table 5: Training Strategies of Models Ashok Basnet (AB) (Basnet, 2021), Ashok Basnet, et. al. (ABE), Aawaj (A) (Basnet et al., 2023), Shruti (S) (* Post processed) (Khadka et al., 2023), Danphe (D), and Munal (M).

to train the vocoder model on NVIDIA V100 GPU. But, we cannot overlook the fact that they trained on a larger dataset. The comparison of training strategies, vocoder, and the MOS scores of different TTS models are shown in Table 5 and Table 4 respectively.

6 Conclusion

The results can be further improved with the help of custom-trained or fine-tuned transliteration models. An alternative solution will be eliminating the need for a transliteration module and training directly on Unicode as the transliteration approach sometimes runs into the risk of incorrect transliteration.

Another area for improvement can be using (MORISE et al., 2016) to extract the ground truth frequencies like mentioned in (Kumar et al., 2023).

We notice some Hindi and Newari accents in the generated audio because of some issues of the transliteration model (Madhani et al., 2022) and SLR43 (Sodimana et al., 2018) dataset respectively. Our model performs relatively poorly with short utterances, as the ILPRLNAB dataset we used to train the model consists of a majority of long utterances. Preparing a phonetically balanced dataset with both the long and short utterances included with a proper Nepali accent will most likely address the issues mentioned above.

In addition, we need to focus on Named Entity Recognition (NER) to identify the non-standard words or semiotic classes like numbers, dates, currencies, etc., and verbalizers to convert text from the written domain to the spoken domain. These issues need to be addressed in the future which is expected to further increase the quality of the synthesized speech.

7 Acknowledgements

We would like to thank the Nepal Association of the Blind (NAB) for providing us with the dataset. We would like to extend our gratitude to all the volunteers who helped evaluate our models.

References

Rohan Badlani, Adrian Łancucki, Kevin J. Shih, Rafael Valle, Wei Ping, and Bryan Catanzaro. 2021. One tts alignment to rule them all.

Roop Bajracharya, Santosh Regmi, Bal Krishna Bal, and Balaram Prasain. 2018. Building a Natural Sounding Text-to-Speech System for the Nepali Language - Research and Development Challenges and Solutions. In *Proc. 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pages 152–156.

Ashok Basnet. 2021. Attention And Wave Net Vocoder Based Nepali Text-To-Speech Synthesis. https://elibrary.tucl.edu.np/handle/123456789/7668.

Ashok Basnet, Basanta Joshi, and Suman Sharma. 2021. Deep Learning Based Voice Conversion Network. In *Proceedings of 10th IOE Graduate Conference*, volume 10, pages 1292 – 1298. Institute of Engineering, Tribhuvan University, Nepal.

Mausam Basnet, Nishan Poudel, Sampanna Dahal, and Sukriti Subedi. 2023. DSpace at Tribhuvan University Central Library (TUCL): AAWAJ : AUGMENTATIVE COMMUNICATION SUPPORT FOR THE VOCALLY IMPAIRED USING NEPALI TEXT-TO-SPEECH — elibrary.tucl.edu.np. https://elibrary.tucl.edu.np/handle/123456789/18839.

Keith Ito and Linda Johnson. 2017. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/.

Supriya Khadka, Ranju G.C., Prabin Paudel, Rahul Shah, and Basanta Joshi. 2023. Nepali text-to-speech synthesis using tacotron2 for melspectrogram generation. In *SIGUL 2023, 2nd Annual Meeting of the Special Interest Group on Under-resourced Languages: a Satellite Workshop of Interspeech 2023*.

Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha. 2018. 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.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis.

Gokul Karthik Kumar, Praveen S V au2, Pratyush Kumar, Mitesh M. Khapra, and Karthik Nandakumar. 2023. Towards building text-to-speech systems for the next billion users.

Yash Madhani, Sushane Parthan, Priyanka A. Bedekar, Ruchi Khapra, Vivek Seshadri, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2022. Aksharantar: Towards building open transliteration tools for the next billion users. *ArXiv*, abs/2205.03018.

Masanori MORISE, Fumiya YOKOMORI, and Kenji OZAWA. 2016. World: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, E99.D(7):1877–1884.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2018. Waveglow: A flow-based generative network for speech synthesis.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J. Mielke, Cibu Johny, Işin Demirşahin, and Keith Hall. 2020. Processing South Asian languages written in the Latin script: the Dakshina dataset. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC)*, pages 2413–2423.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions.

Kevin J. Shih, Rafael Valle, Rohan Badlani, Adrian Lancucki, Wei Ping, and Bryan Catanzaro. 2021. Rad-tts: Parallel flow-based TTS with robust alignment learning and diverse synthesis. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*.

Keshan Sodimana, Knot Pipatsrisawat, Linne Ha, Martin Jansche, Oddur Kjartansson, Pasindu De Silva, and Supheakmungkol Sarin. 2018. 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.

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

Adrian Łańcucki. 2021. Fastpitch: Parallel text-to-speech with pitch prediction.