# End-to-End Speech Synthesis for Indian Languages

*Anusha Prakash*

Indian Institute of Technology Madras, Chennai, India

anushaprakash90@gmail.com

## 1. Introduction

Conventional training of text-to-speech (TTS) synthesisers involves quite some laborious data processing prior to training. It requires separate feature engineering, linguistic expertise (especially in terms of parsing text into their sub-word units), robust alignment of speech data, etc. Attention-based sequence to sequence (seq2seq) models have been quite successful for speech synthesis. The end-to-end speech framework has made things very easy as TTSes can be directly built using <text, audio> pairs [1, 2, 3].

The focus of the thesis is on training end-to-end speech synthesisers for Indian languages. There are 1652 languages in India with 22 official languages written in different scripts. Indian languages are considered to be low resource due to the paucity of (annotated) speech data. To build a high-quality end-to-end seq2seq model for synthesis, tens of hours of data are required [1, 2, 4]. The problem now is to train a good quality TTS in a low resource scenario for multiple Indian languages.

Although the end-to-end framework has made system building easier, there are few issues while dealing with Indian languages. Two major issues are considered in this thesis. The first problem is of different scripts across languages and the second one is of longer utterances. Both these problems and their possible solutions are discussed in the subsequent sections. In all the experiments, ESPNet's implementation [5] of Tacotron2 [2] is used as the end-to-end synthesiser. About 5 hours of monolingual data [6] is used for training.

## 2. Towards multilingual speech synthesis

Tacotron2 takes character sequences as input and generates mel-spectrograms. A WaveNet vocoder auto-regressively conditions on the mel-spectrograms and produces the speech output sample-wise. Character sequences are extracted from the text based on Unicode values. A character map is then prepared, wherein, Unicode values present in the data are mapped to a set of unique tokens. When multiple languages with different scripts are pooled together, the size of the character map becomes large. Indian languages share many similar features. By exploiting these similarities, a multi-lingual character map (MLCM) is proposed. Characters across languages sharing similar acoustic features are mapped to the same token. This results in a compact character map with only 68 unique tokens. The MLCM has been prepared for 13 major Indian languages spanning 8 different scripts. A subset of the cross-lingual mapping is given in Table 1.

As seen from Table 1, pronunciations "aa" and "i" have two rows corresponding to vowels and vowel modifiers. Vowels and corresponding vowel modifiers have different Unicode values in a script, even though they represent the same sound. A pairwise comparison (PC) test is performed to compare two systems– one with vowels and corresponding vowel modifiers mapped together and one without mapping. As seen from Table 2, the system with mapping is preferred. This mapping is incorporated in MLCM.

Table 1: *Examples of cross-lingual mapping in MLCM. Pronunciations are in terms of common label set representation (CLS) [7].*

| Pronuncation in CLS | Bengali | Devanagari | Gujarati | Kannada | Malayalam | Odiya | Tamil | Telugu |
|---|---|---|---|---|---|---|---|---|
| a | অ | अ | અ | ಅ | അ | ଅ | அ | అ |
| aa | আ | आ | આ | ಆ | ആ | ଆ | ஆ | ఆ |
| | া | া | ા | ಾ | ാ | ା | ா | ా |
| i | ই | इ | ઇ | ಇ | ഇ | ଇ | இ | ఇ |
| | ি | ি | િ | ಿ | ി | ି | ி | ి |
| ka | ক | क | ક | ಕ | ക | କ | க | క |
| ga | গ | ग | ગ | ಗ | ഗ | ଗ | - | గ |
| zha | - | - | - | - | ഴ | - | ழ | - |

Table 2: *PC test results (Malayalam female data) - with and without mapping vowels and vowel modifiers (preference in %)*

| Without Mapping | With Mapping | Equal |
|---|---|---|
| 30 | 51 | 19 |

In monolingual experiments, two systems are trained for each language with– (1) language-specific character map (2) MLCM. The former is a subset of MLCM. A degradation mean opinion score (DMOS) test is conducted to evaluate the systems. As expected, the system using MLCM results in a graceful degradation compared to that using the language-specific character map (Table 3).

Table 3: *DMOS scores - monolingual data*

| Language | Language-specific | MLCM |
|---|---|---|
| Bengali | 3.60 | 3.37 |
| Hindi | 2.88 | 2.86 |
| Malayalam | 3.13 | 3.07 |

Experiments are also conducted by pooling data across languages. Two flavours of mixing are considered– (1) Bengali+Hindi (2) Malayalam+Tamil. For each flavour, two systems are trained– (a) with a language-specific character map (one containing characters specific to those scripts alone) (b) MLCM. In the language-specific case too, characters that sound the same across the pooled languages are mapped together. Here too, (b) is a subset of (a). Only monolingual test sentences are synthesised for the DMOS tests. As expected, the system using MLCM results in a graceful degradation compared to that using the language-specific character map (Table 5).

Although there is a slight degradation in the performance of systems with MLCM compared to those using language-specific character-maps, the advantage of MLCM is that it can

Table 4: *Comparison of training time across different systems*

| Language | Training time per epoch (min) | | | | |
|---|---|---|---|---|---|
| | Sentence-based | Phrase-based: $T(sil)$ | | | Average time per epoch for phrases (min) |
| | | 100 msec | 200 msec | 300 msec | |
| Hindi (M) | 5.36 | 2.18 | 2.43 | 2.18 | 2.6 |
| Tamil (F) | 6.23 | 2.5 | 3.52 | 2.3 | 2.77 |

Table 5: *DMOS scores - pooled data*

| Language | Language-specific | MLCM |
|---|---|---|
| Bengali | 3.37 | 3.27 |
| Hindi | 3.05 | 2.81 |
| Malayalam | 3.22 | 3.05 |
| Tamil | 3.12 | 2.88 |

be easily extended to accommodate new Indian scripts. This is especially useful in a multilingual scenario. This work has been extended to include phone-based representations and has been accepted for presentation in Speech Synthesis Workshop (SSW), 2019.

## 3. Inter-pausal phrase-based approach

In general, Indian language texts are longer compared to those in English. Traditional Indian language texts did not have the concept of punctuation marks, including commas, full stops, and spaces between words. Indian language utterances are made up of a sequence of phrases rather than a sequence of sentences.

It has been observed that for longer utterances, seq2seq models do not perform well. Also, for very long utterances, out-of-memory (OOM) issues are encountered. Alignment between linguistic and acoustic features is learnt more robustly for shorter sequences [8]. Hence, it is more effective to splice the utterances into smaller segments for training. For Indian languages, phrases as segments are a good design choice.

Studies and experiments have been carried out by [9, 10] in the context of phrase-based TTSes. In [10], sentence level data is spliced at the phrase-level and a TTS is trained on the phrased data. During synthesis, the test sentence is split into phrases by considering commas as phrase boundaries. Individual phrases are synthesised and concatenated together to give the sentence-level output. Experiments have been carried out in the hidden Markov model (HMM) based domain. The experiments in this thesis are carried out in the end-to-end framework. Also, experiments are carried out by considering different thresholds for phrasing the data.

Two languages are considered in the experiments– Hindi (male) and Tamil (female), which belong to the Indo-Aryan and Dravidian language families, respectively. A hybrid hidden Markov model-deep neural network (HMM-DNN) technique [11] is used to align the sentence level data. The aligned data is split into phrases if the intra-sentential silence region is greater than a certain threshold, termed as $T(sil)$. Experiments are carried out by considering $T(sil)$= 100 msec, 200 msec, 300 msec. Phrase-based and sentence-based systems are evaluated using a pairwise comparison (PC) test. For evaluation, the phrase-based systems with $T(sil)$=300 msec are considered. Results of the PC tests are given in Table 6. For Hindi, the phrase-based system out-performs the sentence based system, while for Tamil both systems are comparable. It is observed that the Tamil recording is slow and has more intermediate pauses compared to Hindi. This indicates that the silence threshold is critical for good synthesis quality. It is also observed that phrase-based systems produce prosodically rich synthesis. A significant advantage of the phrase-based approach is a speed up in training time by more than 50% (Table 4).

Table 6: *Results of pair comparison tests (preference in %)*

| Language | Phrase-based | Sentence-based | Equal |
|---|---|---|---|
| Hindi (M) | 50% | 20% | 30% |
| Tamil (F) | 39% | 36% | 25% |

## 4. Future plans and road map for the thesis

The above work are in the preliminary stages of experimentation. Future directions for research are summarised below:

- Experiments have been carried out using Tacotron2 architecture and Griffin-Lim algorithm as the vocoder as they allow for faster experimentation [12]. The plan is to replace this with the WaveNet vocoder for better speech quality [2].
- Building average voices and adapting it to languages that have a low amount of speech data. The ultimate goal would be built TTSes for languages with no written script.
- Further explore the area of phrase-based systems for different languages and study the effect of varying phrase thresholds on the synthesis speech quality.

## 5. Acknowledgements

I would like to thank my advisors, Dr. S. Umesh and Dr. Hema A. Murthy, for their constant guidance and discussions. I would also like to thank my colleague, Anju Leela Thomas, with whom I have collaborated with for my research work.

## 6. References

[1] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *INTERSPEECH*, 2017, pp. 4006–4010.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R.-S. Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783.

[3] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel Wave Generation in End-to-End Text-to-Speech," in *International*

*Conference on Learning Representations (ICLR)*, 2019. [Online]. Available: https://arxiv.org/abs/1807.07281

[4] Y. Yasuda, X. Wang, S. Takaki, and J. Yamagishi, "Investigation of enhanced Tacotron text-to-speech synthesis systems with self-attention for pitch accent language," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019. [Online]. Available: https://arxiv.org/abs/1810.11960

[5] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *Interspeech*, 2018, pp. 2207–2211.

[6] A. Baby, A. L. Thomas, N. N. L, and H. A. Murthy, "Resources for Indian languages," in *Community-based Building of Language Resources (International Conference on Text, Speech and Dialogue)*, 2016, pp. 37–43.

[7] B. Ramani, S. Lilly Christina, G. Anushiya Rachel, V. Sherlin Solomi, M. K. Nandwana, A. Prakash, S. Aswin Shanmugam, R. Krishnan, S. Kishore, K. Samudravijaya, P. Vijayalakshmi, T. Nagarajan, and H. A. Murthy, "A common attribute based unified HTS framework for speech synthesis in Indian languages," in *Speech Synthesis Workshop (SSW)*, 2013, pp. 291–296.

[8] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 577–585. [Online]. Available: http://papers.nips.cc/paper/5847-attention-based-models-for-speech-recognition.pdf

[9] J. J. Prakash and H. A. Murthy, "An analysis of the distribution of syllables in prosodic phrases of stress-timed and syllable-timed languages," in *Speech Prosody*, 2016, pp. 49–53.

[10] J. J. Prakash and H. Murthy, "Analysis of Inter-Pausal Units in Indian Languages and its Application to Text-to-Speech Synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. PP, pp. 1–1, 06 2019.

[11] A. Baby, J. J. Prakash, R. Vignesh, and H. A. Murthy, "Deep Learning Techniques in Tandem with Signal Processing Cues for Phonetic Segmentation for Text to Speech Synthesis in Indian Languages," in *INTERSPEECH*, 2017, pp. 3817–3821.

[12] D. W. Griffin, Jae, S. Lim, and S. Member, "Signal estimation from modified short-time Fourier transform," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 32, no. 2, pp. 236–243, 1984.