Exploring the Role of Language Families for Building Indic Speech Synthesisers

Anusha Prakash, *Graduate Student Member, IEEE* , Hema A Murthy, *Senior Member, IEEE*

***Abstract*—Building end-to-end speech synthesisers for Indian languages is challenging, given the lack of adequate clean training data and multiple grapheme representations across languages. This work explores the importance of training multilingual and multi-speaker text-to-speech (TTS) systems based on *language families*. The objective is to exploit the phonotactic properties of language families, where small amounts of accurately transcribed data across languages can be pooled together to train TTS systems. These systems can then be adapted to new languages belonging to the same family in extremely low-resource scenarios.**

TTS systems are trained separately for Indo-Aryan and Dravidian language families, and their performance is compared to that of a combined Indo-Aryan+Dravidian voice. We also investigate the amount of training data required for a language in a multilingual setting. Same-family and cross-family synthesis and adaptation to unseen languages are analysed. The analyses show that language family-wise training of Indic systems is the way forward for the Indian subcontinent, where a large number of languages are spoken.

***Index Terms*—end-to-end speech synthesis, Indian languages, language families, low-resource**

I. Introduction

WITH the advent of neural network-based end-to-end (E2E) approaches, training a text-to-speech (TTS) synthesiser has become easier when a large amount of data is available for a language [1], [2]. Systems can be trained quickly using accurate <text, audio> pairs aligned at the sentence level. However, building E2E synthesisers for Indian languages is still a challenge due to the following reasons:

1) India has a wide linguistic diversity, with about 1369 rationalised languages and dialects [3]. Of these, 121 languages are spoken by more than 10,000 people in each language. There are 23 official languages, including English. Building a TTS synthesiser for each language from scratch is difficult, given so many languages.
2) There is a lack of accurately transcribed data for training, which is crucial for a TTS system. This is a bottleneck, especially in the E2E framework, which requires tens of hours of training data to produce high-quality speech [4].
3) There are about 13 scripts that are used for Indian languages. This leads to a significant increase in vocabulary size in a multilingual context.

Anusha Prakash is with the Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai-600036, India (email: anushaprakash@smail.iitm.ac.in).

Hema A Murthy is with the Department of Computer Science & Engineering, Indian Institute of Technology Madras, Chennai-600036, India (email: hema@cse.iitm.ac.in).

Most Indian languages can be broadly classified into two language families—Indo-Aryan and Dravidian. In [5], datasets of languages belonging to the same language family were combined for training. This kind of pooling collectively increases the amount of data, with the added advantage of capturing a wide variety of contexts. A multilanguage character map (MLCM) [6] and a common label set (CLS) [7] for Indian languages were developed to reduce the vocabulary size. Speaker embedding in terms of x-vector [8], [9] was also included during training, primarily for speaker selection during synthesis. Native and cross-lingual syntheses were performed. These systems were then adapted to limited data of a new speaker to synthesise the audio in that speaker's voice.

The current work extensively studies the role of language families in training Indic systems, especially when resources are scarce. Most studies in the literature use a large amount of training data (ranging from 25-1250 hours for monolingual speech synthesis), or at least a pre-trained model that has been trained with a large amount of data. In the current work, we train an initial TTS system in a low-resource scenario (in terms of the amount of clean data and speaker coverage). We use a maximum of 20 hours of multispeaker multilingual data (one speaker for each language). The experiments performed in this work attempt to answer the following questions:

- Is it possible to achieve a good-quality TTS system in such a scenario?
- Can we reduce the compute power as a consequence of using less data without significant degradation in speech quality?
- Suppose we have to train a TTS system for a new language with limited data; what are the best strategies that can be adopted for system building?

The studies in this work attempt to answer these questions by focusing on the role of language families in training TTS systems. We build upon the work in [5] and systematically analyse different scenarios. The novelty of this work is highlighted here:

- This work is one of the first attempts to study the *importance of language families* in the context of speech synthesis.

- We compare language family-specific Indo-Aryan (IA) and Dravidian (Dr) models with a combined Indo-Aryan+Dravidian (IA+Dr) system.
- We also assess the performance of models trained in *data-stressed situations*. We reduce the training data used per language in the multilingual voice.
- *Zero-shot synthesis*: We study the effect of *same language family and cross-family synthesis* for an unseen language.
- Given a limited amount of data for an unseen language, we study *different scenarios of adaptation*—same family, cross-family and IA+Dr adaptation.
- We highlight the differences between Indo-Aryan and Dravidian languages and quantify the differences by *phonotactic*[1] *analysis* using byte-pair encoding (BPE) [10], [11] and language modelling.

In this work, multilingual and multispeaker voices (referred to as generic voices) are trained using single-speaker data per language. Most of the experiments do not use speaker embeddings as we do not aim to synthesise speech in a particular speaker's voice. The primary objective is to preserve language characteristics in terms of phonotactics rather than speaker characteristics. For completeness, the results of systems trained with speaker embeddings are presented in the supplementary material (Sections S1, S2 and S4). The performance of systems is quantified using objective and subjective measures and supported by qualitative analysis in terms of informal listening tests.

The rest of the paper is organised as follows. Section II highlights the motivation to train systems based on language families. The literature on related work is discussed in Section III. Sections IV and V present the experiments and analysis of the various TTS systems. Observations are summarised in Section VI, and the phonotactic analyses of languages is presented. The work is concluded in Section VII.

II. Motivation

The written scripts of Indian languages can be traced to the *Brahmi* script. Although different Indian languages may have different grapheme representations, they share a common set of sounds. Most languages have about 11–15 vowels and 33–35 consonants, except Tamil, which has representations for only 23 consonants. Despite a common set of phones, phonotactics across languages varies. Indian languages have simple phone clusters and are *akshara*-based [12]. In comparison, English has complex phone clusters such as *twelfth* and *strength*. Phone clusters such as *sr* and *ph* (aspirated *p*), rarely occur in English. Similarly, phone clusters such as *ion* and *ous* are quite rare in Indian languages [13]. Unlike English, Indian languages are replete with geminates [14].

Phonotactic differences are especially evident across language families. Most Indo-Aryan languages are characterised by *schwa* deletion, which is the absence of the inherent short vowel *a* [15]. *Agglutination*, which refers to the phenomenon of combining multiple words, is very common in Dravidian languages [16]. Language-specific phones also contribute to these phonotactic differences. Dravidian languages have many liquids and distinguish between alveolar, dental and retroflex places of articulation. Dravidian languages are also characterised by their lack of distinction between aspirated and unaspirated stop consonants. Telugu, Malayalam and Kannada scripts have representations for aspirated stop consonants primarily to accommodate the use of borrowed words from Sanskrit. Motivated by these differences, this work explores the effectiveness of training systems based on language families.

[1]Phonotactics is the sequence of phones that is allowed in a language.

III. Related Work

There have been previous attempts to train generic voice models from different perspectives—polyglot synthesis [17]–[28], code-mixing[2] [23], [26], [29]–[36], cross-lingual voice conversion [37], [38], and data augmentation [21], [22], [25], [26], [28], [39]–[44]. Polyglot synthesis aims to synthesise texts of multiple languages in the voice of a single speaker. Cross-lingual voice conversion deals with synthesising the linguistic content of a source speaker in the voice of a target speaker. Data augmentation aims to increase the collective data for training by pooling data across languages. The perspective of the current work can be considered to be but not limited to a combination of polyglot synthesis (using multispeaker data) and data augmentation. As a consequence of multilingual training followed by adaptation, the work can be extended to other perspectives.

An effective solution to multilingual training is to collect data of a single person speaking multiple languages, as performed in [17], [22], [29], [31], [37], [45]–[48]. This is especially essential in the unit selection synthesis (USS) paradigm [17], [29], where waveforms are directly concatenated. Collecting single-speaker multilingual speech data may not always be feasible, and extension to new languages becomes restrictive. Most studies address this bottleneck by combining several monolingual databases recorded by different speakers. One study attempts to generate a polyglot database in a target voice by cross-lingual voice conversion [23]. Similarly, in [36], bilingual data is generated using voice conversion and the training data is augmented to build a TTS system capable of code-mixing. [39] trains a USS synthesiser utilising a mixture of monolingual corpora and then transforms the synthesised utterances to a target voice. A popular approach in the hidden Markov model (HMM) based TTS paradigm is to map/share attributes across languages, such as phone mapping [18]–[20], [31], and sharing of HMM states [30]. In [21], speaker and language-specific characteristics are modelled using separate transforms.

[2]Code-switching is the alternation between multiple languages in a single conversation. Code-switching is an inter-sentential phenomenon, while code-mixing is more of an intra-sentential phenomenon.

In the domain of neural networks, speaker-independent and language-independent layers of a deep neural network (DNN) are shared. These layers serve as a bridge between speaker-specific and language-specific layers [22]. Similarly, [40] trains a multilingual bidirectional long short-term memory (BLSTM) neural network in which the hidden layers across different languages are shared. In contrast, the input and output layers are considered to be language-dependent. [41] trains an LSTM-RNN (recurrent neural network) based system, wherein language and speaker variations are modelled using cluster adaptive training and speaker-dependent layers, respectively. [43] proposes a multilingual phoneme inventory and trains a multilingual and multispeaker LSTM-RNN model.

Recent literature on multilingual training is mainly in the E2E framework. The text processing module or the text encoder is modified to enable multilingual training. [32] explores two types of text encoders—(a) a single multilingual encoder with language embedding and (b) a separate encoder for each language. In [49], the text is represented as a sequence of bytes, thus rendering the text encoder language-independent. In a few experiments, international phonetic alphabet (IPA) based features, more widely known as phonological features, are used in multilingual training [27], [35], [50].

As TTS systems are trained in multispeaker and multilingual settings, additional embeddings such as speaker and language embeddings are included during training [24], [28], [33]. This enables the synthesis of any language in any speaker's voice. To improve cross-lingual synthesis, a popular technique is to disentangle speaker information and linguistic content [26], [51]–[53]. In [26] and [51], an adversarial loss is included to suppress speaker-dependent information. [52] uses domain adaptation objective to obtain language-independent speaker embedding and inter-speaker perceptual similarity to train a speaker encoder. [53] attempts to disentangle speaker and spoken content by minimizing the mutual information between them. In this context, [38] summarises various techniques used for the cross-lingual voice conversion task in the voice conversion challenge 2020.

On average, multilingual TTS systems are comparable to single/multi-speaker monolingual systems for synthesising text in the same language. Multilingual systems are also extended to extrinsic languages, as shown in [20], [42], [43], [46]. Further, multilingual systems are adapted to a (new) language or speaker [18]–[21], [23], [25], [28], [33], [41], [46], [50], [54]–[58].

Experiments in [51] and [59] aim to remove foreign accents in cross-lingual synthesis. However, a non-native accent need not be an undesirable entity [19] and can be considered to mimic the real-world scenario. Hence, no attempts have been made to remove the accent in the synthesised speech in the current work.

In the context of multilingual E2E training for Indian languages, [54] trains convolutional attention-based TTS with language, speaker and gender embeddings. In [56], pre-training strategies are explored between source and target languages, which enable the training of multilingual voices with a reduced amount of data. In [58], byte inputs are mapped to spectrograms and experiments are performed with 40+ languages, including Hindi, Tamil and Telugu.

Most of the above literature uses a huge amount of data (ranging from tens to hundreds of hours) for training generic voices. For example, [58] uses close to 900 hours for training generic systems. In the current work, we use a maximum of 20 hours of accurately transcribed data for training an initial generic voice. Most importantly, in contrast to the above-presented literature, we explore the role of language families in system training. Although the TTS systems in [5] are trained based on language families, the relevance of training them in this manner is not explored. A recent study [60] observes that language family classification may not be an effective basis for choosing (source) languages for training a generic TTS. However, our own experiences with multilingual TTS systems and observations in [18] find that the intelligibility of synthesised speech depends on the similarity between any target language and source language(s).

During the review process of this paper, a recent and parallel work has performed extensive studies on how various factors in training affect polyglot synthesis quality [61]. Specifically, the focus is on factors such as gender, speaker composition, and language family affiliation. Analysis of unseen language synthesis is performed by adding language variants belonging to the same and different language families in the training data. The paper concludes that adding languages to the training data closer to a target language is better than adding a dissimilar language (similar to observations drawn in [18]).

Another motivation for undertaking this study is the lack of comprehensive literature on building E2E TTS systems for Indian languages. Although there is an impetus in building E2E synthesis systems, with new architectures and techniques being developed, there is very little work on addressing the challenges specific to the Indian context. There is a need to evaluate the problem from a low-resource scenario and a multilingual perspective. The authors hope this study will bridge the gap and help future researchers.

IV. Training Indic TTS synthesisers

This section describes the modules in building generic TTS systems and their adaptation to new speakers (and languages). Details about the datasets, the representation of multiple scripts and the E2E training of systems are presented.

A. Datasets

The datasets used in this work are part of the Indic TTS database [62][3]. Each dataset consists of speech waveforms and the corresponding text in UTF-8. Details of

[3]Link to the IndicTTS database: www.iitm.ac.in/donlab/tts/database.php

TABLE I: Details of the datasets used. Examples of phone-based (CLS) and character-based (MLCM) representations are given. "F" and "M" refer to female and male datasets, respectively. *Unseen* languages are not used for training generic systems.

Language family	Language	Script	Example	CLS representation	Characters (MLCM tokens)	Gender (duration - hrs)
Indo-Aryan	Bengali	Bangla	বাংলা	b-A-q-l-A	ব - া - ং - ল - া (46-9-6-52-9)	F(5), M(5)
	Hindi	Devanagari	हिन्दी	h-i-n-d-I	ह - ि - न - ् - द - ी (59-10-42-68-40-11)	F(5), M(5)
	Odia	Odia	ଓଡିଆ	o-ɔ-i-A	ଓ - ଡ - ି - ଆ (21-35-10-9)	F(5), M(5)
	Rajasthani	Devanagari	राजस्थानी	r-A-j-a-s-th-A-n-I	र - ा - ज - स - ् - थ - ा - न - ी (50-9-30-58-68-39-9-42-11)	F(5), M(5)
	Gujarati (*unseen*)	Gujarati	ગુજરાતી	g-u-j-r-A-t-I	ગ - ુ - જ - ર - ા - ત - ી (25-12-30-50-9-38-11)	F(0.5), M(0.5)
Dravidian	Kannada	Kannada	ಕನ್ನಡ	k-a-n-n-a-ɔ-a	ಕ - ನ - ್ ನ - ಡ (23-42-68-42-35)	F(5), M(5)
	Malayalam	Malayalam	മലയാളം	m-a-l-a-y-A-ɯ-a-q	മ - ല - യ - ാ - ള - ം (48-52-49-9-53-6)	F(5), M(5)
	Telugu	Telugu	తెలుగు	t-e-l-u-g-u	త - ె - ల - ు - గ - ు (38-16-52-12-25-12)	F(5), M(5)
	Tamil (*unseen*)	Tamil	தமிழ்	t-a-m-i-Z	த - ம - ி - ழ - ் (38-48-10-54-68)	F(0.5), M(0.5)

the datasets are given in Table I. Totally 9 languages are considered—5 belonging to the Indo-Aryan language family (Bengali, Gujarati, Hindi, Odia, Rajasthani), and 4 belonging to the Dravidian language family (Kannada, Malayalam. Tamil, Telugu). Each language has its own unique script, except Hindi and Rajasthani, which share the *Devanagari* script. It is to be noted that only 2 speakers (1 female and 1 male) are considered in each language. Gujarati and Tamil are considered unseen languages as they are not used in training the generic voices. Since attention in encoder-decoder architecture is not learnt effectively on long training utterances, utterances whose duration is less than or equal to 15 seconds are considered. The amount of training data used from each dataset is given in Table I.

B. Text representation

Average voice models are trained in a multilingual context by combining data across languages and training a single network. Since each language has its own unique script, the combined vocabulary size becomes large, leading to poor training of the TTS system. Hence, the texts in different native scripts are mapped to a common representation. The multi-language character map (MLCM) [6] and the common label set (CLS) representations [6], [7] are used. Both MLCM and CLS operate on the principle that acoustically similar subword units across languages are given a common representation. While MLCM is designed for character-based representation, CLS is used for phone-based representation. In the MLCM representation, the text is first split into its constituent characters and then mapped to a set of token numbers using MLCM. To obtain the phone-based CLS representation, the unified parser for Indian languages [63] is used. An example of the UTF-8 word in the corresponding language and its character and phone-based representations is given in Table I. Both MLCM and CLS aid in training systems with a compact representation, with 68–72 tokens rather than 250+ tokens when 4 languages are pooled, or 500+ tokens when 8 languages are pooled.

C. Training E2E systems

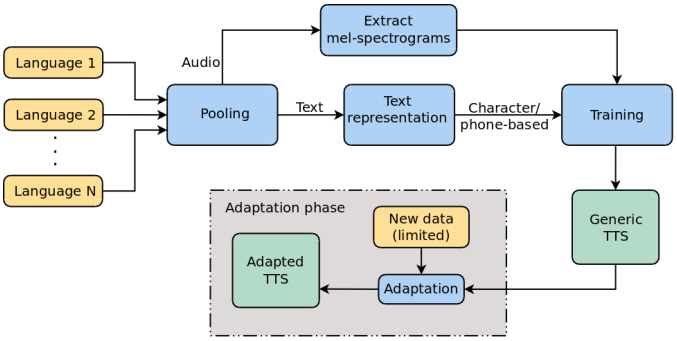

Fig. 1: Flowchart illustrating the training phase of generic voices. The grey box corresponds to the adaptation phase.

An illustration of the training process is given in Figure 1. Data from different languages are combined, and a single neural network is trained. The training phase is divided into two parts—(1) text-to-mel spectrogram mapping based on the Tacotron2 architecture [2]. (2) mel-spectrogram to speech waveform generation using a WaveGlow vocoder [64]. The likelihood P(mel-spectrogram frames|text) is parameterised using an encoder and a decoder with attention. The text is processed into a set of tokens and embedded into a continuous vector. The embeddings are passed through a series of convolution layers to capture the long-term context in the input. The output is then presented to a BLSTM layer to generate the encoded features. The attention network summarises the encoded features into a fixed-length context vector. An auto-regressive decoder predicts the mel-spectrogram frame at each time step. The WaveGlow vocoder is used to generate the time-domain speech signal by conditioning on the mel-spectrogram [64].

D. Adaptation

To improve the synthesis quality of new (unseen) languages, generic voices are adapted using limited amounts of accurately transcribed data from the target language. Specifically, the network parameters of the generic network are fine-tuned on the adaptation data. In [5], different amounts of adaptation data were considered—30 minutes, 15 minutes and 7 minutes. In the current work, we further reduce the amount of adaptation data (3 minutes, 1 minute) and test the limits of transferability in extreme resource-scarce scenarios. The adaptation process is shown in Figure 1. We also investigate the effect of adapting from different generic systems—(1) of the same language family, (2) of a different language family, and (3) combined Indo-Aryan+Dravidian voice.

V. Exploring the relevance of language families for system building

This section gives an overview of the language family-based analysis carried out. Different evaluation metrics are considered to assess the various systems' performance.

A. Experiments

TTS systems are built using ESPNet's implementation [65] of Tacotron2. Training and validation sets are in the ratio 9:1. The validation set is a representative mixture of all the languages considered for training. Different generic voices are trained for male and female datasets to avoid the issue of gender in synthesis. The configuration of the encoder-decoder network used in the experiments is given in Table II. Nvidia's WaveGlow implementation is used for speech reconstruction [64]. WaveGlow models are trained for Indo-Aryan and Dravidian data by fine-tuning a pre-trained ljspeech [66] WaveGlow model for $10,000$ steps. Speaker-dependent models are also trained for each target speaker, but there is no observable difference in synthesis quality across speaker-independent and speaker-specific WaveGlow models.

TABLE II: Tacotron2: Network configuration

Name	Value
Character embedding dimension	512
Encoder layers	1
Encoder units	512
Decoder layers	2
Decoder units	1024
Attention dimension	128

B. Systems built

Various combinations of voices are trained by considering the entities mentioned in Table III. To train the Indo-Aryan (IA) voice, 5 hours each of Bengali, Hindi, Odia and Rajasthani data are combined. The Dravidian (Dr) voice is trained using 5 hours each of Kannada, Malayalam and Telugu data. IA and Dr voices trained collectively with 20 hours and 15 hours of data, respectively, are called "full" voices. Monolingual voices built using individual language datasets (5 hours in duration—Table I) are considered baseline systems. Single-family voices are also trained with only 5 hours of collective data, wherein each constituent language has an equal contribution. This is a data-stressed situation, and these voices are referred to with a "5hrs" tag. The idea is to compare the performance of single-family and monolingual TTS systems trained with the same duration. Further, Indo-Aryan+Dravidian (IA+Dr) voices are also trained for comparison. These voices are trained with a total of 35 hours of data. The above systems are also compared based on their text representation—character-based (MLCM) and phone-based (CLS). Overall, 16 single language family voices and 4 IA+Dr combined voices are built, considering male and female datasets.

These multilingual systems are adapted to unseen (new) languages. As mentioned in Section IV-D, three types of adaptation are carried out:

- Same-family adaptation: IA and Dr voices are adapted to Gujarati and Tamil, respectively, with varying amounts of data (20 adapted systems in total).
- Cross-family adaptation: Dr and IA voices are adapted to Gujarati and Tamil, respectively (4 adapted systems in total).
- Combined IA+Dr voice adaptation: IA+Dr voice is adapted to Gujarati and Tamil (4 adapted systems in total).

In addition to the systems mentioned in Table III, the following systems are also trained:

- Multilingual and adapted systems with x-vectors as speaker embedding.
- A single IA+Dr voice with x-vector, combining male and female datasets.

x-vectors are extracted from audio files using a pre-trained time-delay neural network (TDNN) [8] and then appended to each encoder state of the Tacotron2 network. Compared to the systems without speaker embedding, the systems with speaker embedding have better speaker stability and improved quality in a few cases. Nevertheless, from a language family-based perspective, results are similar with/without speaker embedding. Hence, the results of these additional systems are presented in the supplementary material (Sections S1, S2 and S4).

C. Test set and evaluation metrics

Held-out sentences not used for training are considered for evaluations. This set does not overlap with the data mentioned in Table I. The test set for each dataset has at least 100 sentences and covers at least 10 occurrences of each phone. The length of the test set ranges from 114 to 178 sentences. Table IV summarises the languages and the number of test sentences considered in each dataset.

The following evaluation metrics are used to analyse the synthesised audio of different TTS systems:

TABLE III: List of systems trained*

Category	Language families	Gender	Types of TTS systems	Amount of adaptation data (in minutes)
Single family	Indo-Aryan (IA) Dravidian (Dr)	Male Female	1. Single family (MLCM, full) 2. Single family (CLS, full) 3. Single family (MLCM, 5hrs) 4. Single family (CLS, 5hrs)	–
Combined	Indo-Aryan+Dravidian	Male Female	1. IA+Dr (MLCM) 2. IA+Dr (CLS)	–
Adapted (same language family)	Gujarati (IA) Tamil (Dr)	Male Female	Best single family (full) voice from the same language family adapted	30, 15, 7, 3, 1
Adapted (cross language family)	Gujarati (Dr) Tamil (IA)	Male Female	Best single family (full) voice from a different language family adapted	7
Adapted (combined IA+Dr)	Gujarati (IA+Dr) Tamil (IA+Dr)	Male Female	IA+Dr (CLS) combined voice adapted	7

*Monolingual baseline systems excluded

TABLE IV: Statistics of test set

Dataset	Gender (no. of test sentences)	Dataset	Gender (no. of test sentences)
Hindi	M (169), F (167)	Kannada	M (178), F (120)
Odia	M (134), F (134)	Malayalam	M (157), F (169)
Rajasthani	M (151), F (149)	Tamil	M (120), F (114)
Gujarati	M (130), F (140)	-	-

1) Mel-cepstral distortion (MCD): MCD is an objective evaluation metric used to measure the distortion in mel-cepstral features of synthesised speech compared to that of the corresponding recorded speech [67]. Dynamic time warping (DTW) is first performed to align the speech signals. A lower average MCD indicates that the TTS system produces less distorted speech.

2) MUSHRA test: MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) is a subjective evaluation metric used to assess the perceptual quality of the synthesised speech [68]. Synthesised utterances (of the same sentences) generated by various TTS systems are presented to the listeners on a single panel. For each panel, the order of systems is randomised. Listeners are asked to rate the quality of the synthesised speech with respect to a reference. The scoring is on a scale of 1–100; a score of "100" indicates that the quality of the synthesised utterance is the same as that of the reference audio.

3) Additional subjective evaluations: Customised subjective evaluations such as intelligibility tests and language verification (LVF) tests are conducted. The tests are detailed in the relevant sections.

4) Additional qualitative observations: In addition to the above formal evaluation methods, informal analysis is also conducted to verify the observations. This includes manual verification, informal listening tests, and feedback on the synthesised audio. Attention plots of the sequence-to-sequence models are also studied.

D. Analysis

The performance of various voices is analysed using the metrics mentioned above[4]. Synthesis using multilingual voices is divided into two categories—(1) synthesis of seen languages and (2) synthesis of unseen languages. Only languages seen during training are synthesised and evaluated in the first scenario. In the second scenario, the text of unseen languages is synthesised. It is to be noted that generic voices have not been fine-tuned for any in-training speakers.

*1) **Analysis of generic systems for seen languages:*** **MCD scores:** Figure 2 shows the MCD scores corresponding to male TTS systems. The x-axis in the plot refers to the language of the native text. Each text is passed through 7 different voices—baseline monolingual TTS voice of that language, four single-family (IA/Dr) and two combined IA+Dr systems built using different text representations.

It is seen from Figure 2 that baseline monolingual systems perform better than generic systems in most cases. Considering only single family (full) systems, character-based representation performs better than phone-based (CLS) representation for Indo-Aryan languages. The reverse is true for Dr (full) voices. The phone-based representation performs best for all languages for single-family (5hrs) systems. The degradation in performance of the phone-based IA (full) system could be a consequence of incorrect grapheme-to-phoneme conversion (mainly *schwa* deletion), which has become more prominent in the full voice than in the 5hrs voice. There is no significant difference between the performances of both IA+Dr systems. Only for Kannada and Malayalam, MCD scores are better for IA+Dr voices compared to monolingual and single-family voices. However, this difference is not very significant, given that IA+Dr voices are trained on a considerable amount of data (35 hours compared to 5/15/20 hours of monolingual or single-family voices). Comparing MCD scores of monolingual TTS systems (with 5 hours of training data) and the best single-family (5hrs) systems, the average relative degradation with respect to the former is only 3.82%. A similar comparison of the best single-family (5hrs) system with the best IA+Dr system indicates an average relative degradation of 3.57% with respect to the latter. This is an encouraging result, given that the IA (5hrs) and Dr (5hrs) voices are trained with only 1.25 hours and 1.67 hours of data per language, respectively.

Similar results are observed for systems trained on female data and systems trained with speaker embedding, as presented in the supplementary material (Sections S1 and S2). The average relative degradation in MCD score

[4]Synthesised samples are available at www.iitm.ac.in/donlab/preview/TTS_language_family/index.html

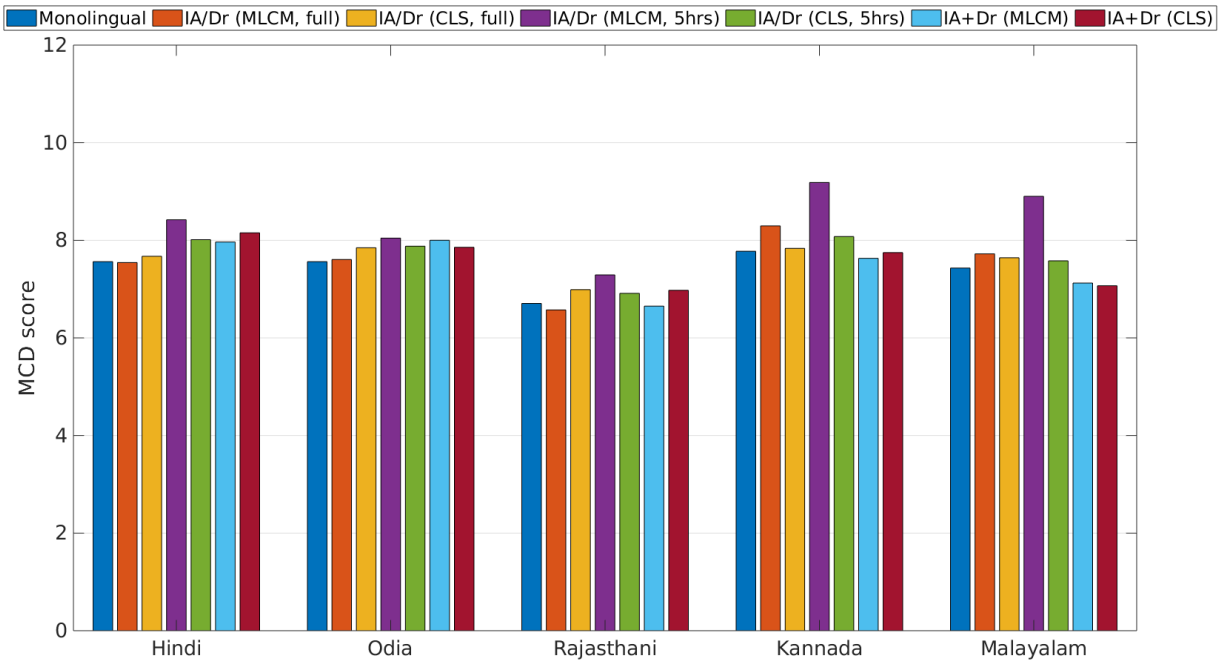

Fig. 2: MCD scores of monolingual text of different languages synthesised by various systems built using male data

of the best single-family (5hrs) voice compared to that of monolingual systems is only 4.5%, and with respect to the best IA+Dr system is 3.14%.

Subjective intelligibility tests: A subjective word error rate (WER) test is carried out to measure the intelligibility of each system. This test is performed only for Hindi and Kannada systems trained using male data. Evaluators are presented with sentences and corresponding audio files synthesised by each system. The evaluators were asked to enter the number of words wrongly pronounced in the synthesised utterances. Although providing the text does bias the participant, this bias is uniform across all systems. For the evaluation, synthesised utterances corresponding to 10 randomly selected sentences were considered. Details of the evaluation and WER across all systems are presented in Table V. The results are more or less similar to the patterns observed for MCD scores. Monolingual TTS and IA+Dr (MLCM) perform the best for Hindi and Kannada, respectively. Single-family (MLCM, 5hrs) systems have the highest WER.

MUSHRA tests: MUSHRA tests are conducted to assess the quality of synthesised utterances across various systems. Based on MCD scores and informal listening tests, IA (MLCM, full) and Dr (CLS, full) voices are the best single-family voices. IA+Dr (CLS) combined voice performs better than IA+Dr (MLCM) voice in most cases. Hence, MUSHRA tests are conducted for these systems, along with corresponding single-family voices in data-stressed situations (IA (MLCM, 5hrs), Dr (CLS, 5hrs)). Monolingual TTS synthesisers are also included for comparison.

Native listeners participated in the evaluations—Hindi (18), Odia (5), Rajasthani (5), Kannada (11), and Malayalam (19). Each listener evaluated a set of 20 audio files in each test, 5 from each system. Figure 3 presents the MUSHRA scores for male voices. In most cases, the synthesis quality of single-family (5hrs) voices is the least, followed by the IA+Dr voice. The performance of single-family (full) and monolingual systems is similar in most cases, except for Hindi and Kannada. For Kannada, multilingual training (particularly IA+Dr) seems to improve synthesis quality compared to monolingual training. Similar results are observed for systems trained on female data, as presented in the supplementary material (Section S3). On average, the relative degradation in MUSHRA score of single-family (5hrs) voices is 12.93% compared to the best system.

*2) **Analysis of generic systems for unseen languages (zero-shot scenario)**:* The scenario of synthesising text from unseen languages (Gujarati and Tamil) is analysed. Only single-family voices are considered here. We aim to study the extent to which language families can affect the synthesis in unseen languages. Two types of cases are explored:

TABLE V: Subjective intelligibility tests: WER of various systems trained using Hindi and Kannada male datasets

Language	Average no. of words per sentence	No. of evaluators	Monolingual	IA/Dr (MLCM, full)	IA/Dr (CLS, full)	IA/Dr (MLCM, 5hrs)	IA/Dr (CLS, 5hrs)	IA+Dr (MLCM)	IA+Dr (CLS)
Hindi	17	13	2.57%	3.58%	6.49%	12.14%	8.98%	4.22%	6.58%
Kannada	12	9	8.67%	9.92%	5.33%	16.67%	7.33%	4.00%	4.33%

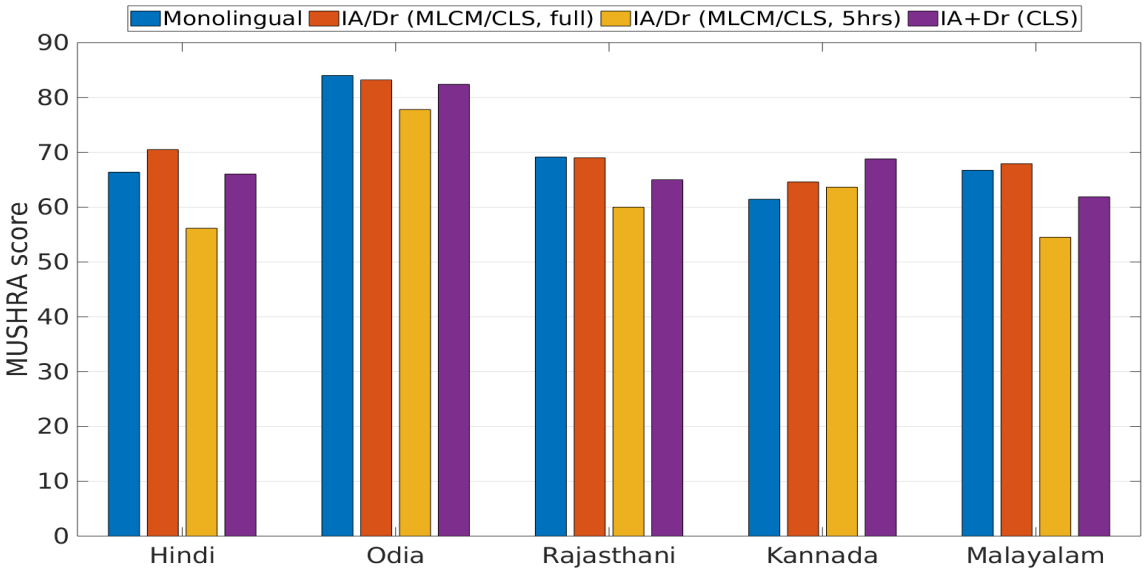

Fig. 3: MUSHRA scores of monolingual text of different languages synthesised by various systems built using male data

1) Same language family synthesis—Gujarati and Tamil texts are synthesised by IA and Dr voices, respectively.
2) Cross-language family synthesis—Tamil and Gujarati texts are synthesised by IA and Dr voices, respectively.

IA (MLCM, full) and Dr (CLS, full) TTS systems are considered as these are the best systems in the respective language families. The unseen language text is passed directly to the single-family voice during synthesis. It is to be noted that although these languages are not used during training, the CLS and MLCM representations can handle them.

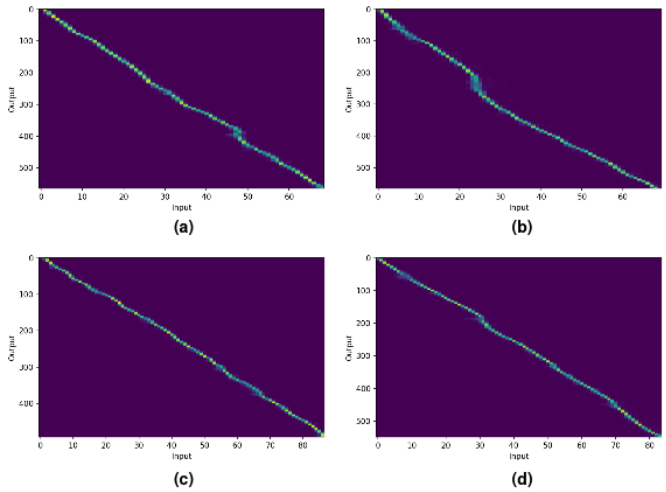

Fig. 4: Attention plots corresponding to (a) Gujarati text synthesised by IA voice (b) Gujarati text synthesised by Dr voice (c) Tamil text synthesised by IA voice (b) Tamil text synthesised by Dr voice.

Attention plots of IA and Dr male TTS systems for sample Gujarati and Tamil texts are shown in Figure 4. The monotonic nature of the plots indicates that the synthesised utterances are reasonably intelligible. However, informal listening tests present a different story. Both cases of synthesis have a non-native accent, which is expected. Same language family synthesis is relatively more intelligible compared to cross-family synthesis. However, the accent in cross-language family synthesis is quite pronounced and impedes intelligibility. Clearly, languages are not only different due to phonotactics but also prosody. Differences in the phone sets between the unseen language and the single-family voice further contribute to this degradation. Evaluating non-native synthesised utterances is not trivial. We have designed a subjective language verification (LVF) test to assess both cases of unseen language synthesis. Details of the LVF test are presented in Section V-D3 along with a comparison with adapted systems.

A note on speaker identity and stability: Since there is only one speaker per language in the training data, the problem of stability of speaker identity is largely avoided. To quantify this, a speaker identification (SID) system is built by combining the training data mentioned in Table I. The idea is to see to what extent the speaker identity of synthesised seen language is in the corresponding seen speaker's voice. On average, this value is 90.5% and 91.8% for language family-specific and IA+Dr voices, respectively. It is also observed that speaker similarity does not necessarily influence speaker identity in synthesis. Details on this are presented in the supplementary material (Section S6). With x-vectors, we can explicitly specify a voice for synthesis.

For unseen languages, the synthesised speech is in the voice of a seen speaker, which varies with the test sentence. As seen in [5], even if we specify the speaker embedding of

the unseen speaker during synthesis, this is not reflected in the speaker identity of the output audio. It is still in the voice of a seen speaker.

*3) **Analysis of adapted systems (same language family)**:* A TTS system built using only a limited amount of data (say, 30 minutes) in an unseen language does not train well. Hence, generic TTS systems are fine-tuned on this limited data to improve unseen language synthesis. Adaptation is performed within the same language family, and the best single-family systems are considered. IA (MLCM, full) and Dr (CLS, full) TTS systems are adapted to Gujarati and Tamil, respectively. We study adaptation with varying amounts of data—30, 15, 7, 3 and 1 minute. Table III shows the combinations of adapted systems from the same language family. Adaptation is performed separately for male and female data. The test set corresponding to these languages in Table IV is used for synthesis. For example, in the case of Gujarati female, we use only 5 utterances (1 minute) for adaptation, but test the system on 140 sentences.

MCD scores: Figure 5 presents the MCD scores of different adapted systems. MCD scores are plotted against the amount of adaptation data used. As the amount of adaptation data reduces, it is seen that the MCD score increases. The system's robustness reduces, resulting in high variance and more edge cases and outliers (as indicated by the + symbol in the plots). The performance of the Gujarati male voice drops significantly with 1 minute of adaptation data (Figure 5 (a)). The performance degrades gracefully with reduced adaptation data for the remaining scenarios.

Language verification (LVF): A novel language verification subjective evaluation metric is developed, where subjects are asked to verify the language of the synthesised output. This is performed only for the unseen languages, Gujarati and Tamil. Evaluators are presented with a set of 24 audio files randomly ordered—(a) 4 original recordings of the language, and (b) 5 audio files each synthesised by IA (MLCM, full), Dr (CLS, full), adapted (1 minute) and adapted (30 minutes) voices. The adapted models belong to the same family adaptation. The adapted models are included to assess to what extent evaluators can verify the language with 1 minute and 30 minutes of clean data.

Evaluators are asked to rate if the audio clip is of the unseen language, disregarding any foreign accent. A 5-point rating scale is used, with the following indications:

- Score 5: sure that the audio clip is of that language.
- Score 3: the audio clip could be of that language.
- Score 1: the audio clip is not at all of that language.

7 and 11 native listeners participated in the Gujarati and Tamil LVF tests, respectively. Table VI presents the results of LVF test. It is seen that with cross-family synthesis, the LVF score is less compared to that of the same family synthesis. This is especially evident with the synthesis of Tamil text using IA voice, for which evaluators have rated that the language is not Tamil. Evaluators have

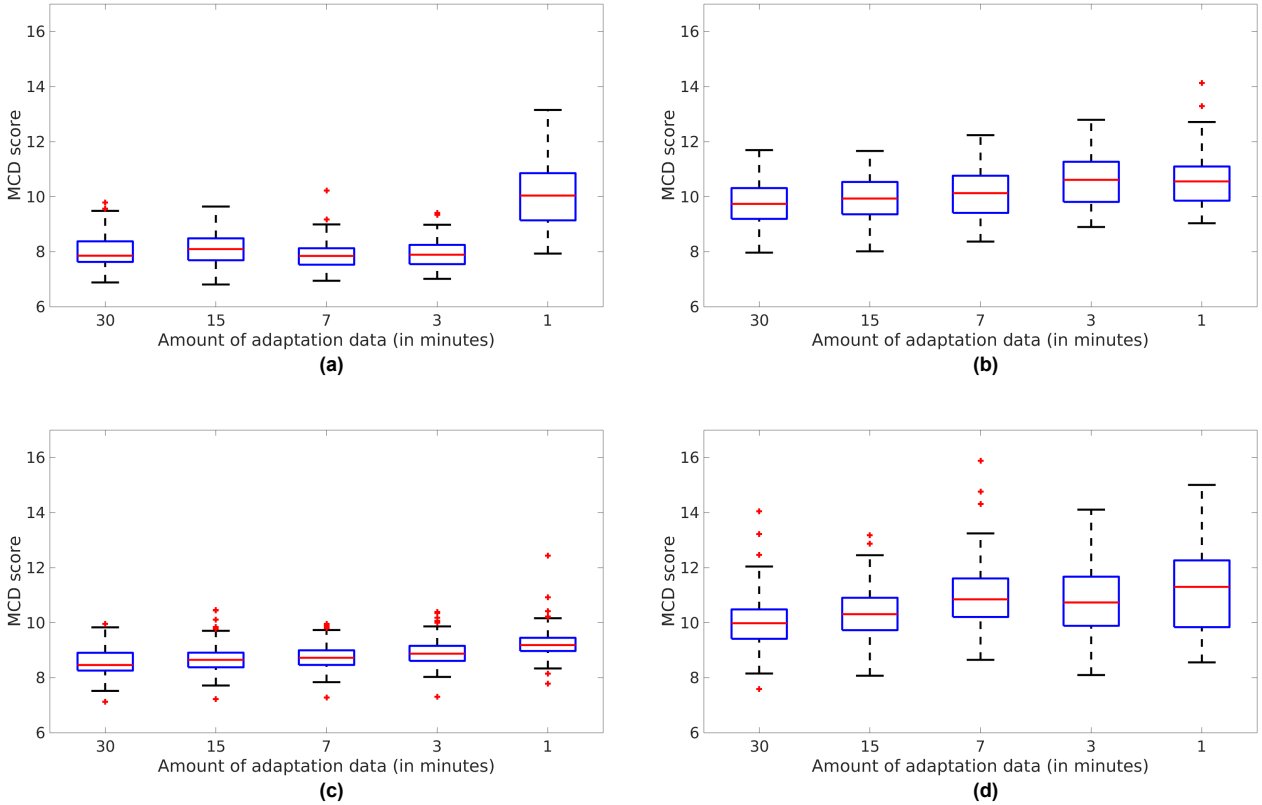

Fig. 5: MCD scores of adapted TTS systems with different amounts of adaptation data: (a) Gujarati (male) (b) Gujarati (female) (c) Tamil (male) (d) Tamil (female).

TABLE VI: Results of language verification test comparing same-family synthesis, cross-family synthesis and adapted systems for unseen languages

TTS system	Gujarati	Tamil
IA TTS	4.69	1.31
Dr TTS	3.29	3.83
Same family adaptation (1 min)	3.21	4.32
Same family adaptation (30 mins)	4.74	4.75

mostly indicated that the language is Gujarati for Gujarati text synthesised using the IA voice. With 1 minute of adaptation data, the LVF score improves over the same family synthesis for Tamil. However, for Gujarati, the score degrades considerably. This is because the quality of the system adapted with 1 minute of Gujarati data is poor (as seen in Figure 5 (a)), and this impacts its language verification. For systems trained with 30 minutes of adaptation data, evaluators are fairly confident that the language of the text is indeed the same. The evaluations are conducted for systems built using male data, and informal evaluations also indicate similar results for female data. These evaluations indicate the importance of same language family synthesis, even when no training data is available for unseen languages.

*4) **Analysis of adapted systems (all scenarios):*** To get a better understanding of how important language families are in the context of adaptation, different generic voices are adapted to Tamil and Gujarati as given in Table III. Three scenarios of adaptation—same language family, cross-language family and IA+Dr adaptation are explored, as elaborated in Section V-B. Only 7 minutes of data from each language is considered for adaptation.

MCD scores: Figure 6 presents the MCD scores of various adapted TTS voices corresponding to Gujarati and Tamil male and female datasets. The x-axis indicates the generic system used for adaptation. It is seen that the MCD scores are slightly higher in cross-language family adaptation compared to the same family adaptation. Informal listening tests indicate that language-specific phones, especially in Tamil, are not pronounced correctly with cross-family adaptation. The performance of IA+Dr voice adaptation and the same family adaptation is almost on par. The difference in performance of same family and cross family adaptation is statistically significant ($p < 0.05$) and that between same family and IA+Dr adaptation is not very significant ($p > 0.05$). This indicates that the same language family voice, trained on a small amount of data, is sufficient for effective adaptation. A combined IA+Dr voice can also be adapted if substantial data is available across language families.

In the adaptation experiments presented here, only the best generic systems are adapted– IA (MLCM, full), Dr (CLS, full), IA+Dr (CLS). The observations on language families still hold even when other generic (MLCM/CLS) models are adapted (Section S4 in the supplementary

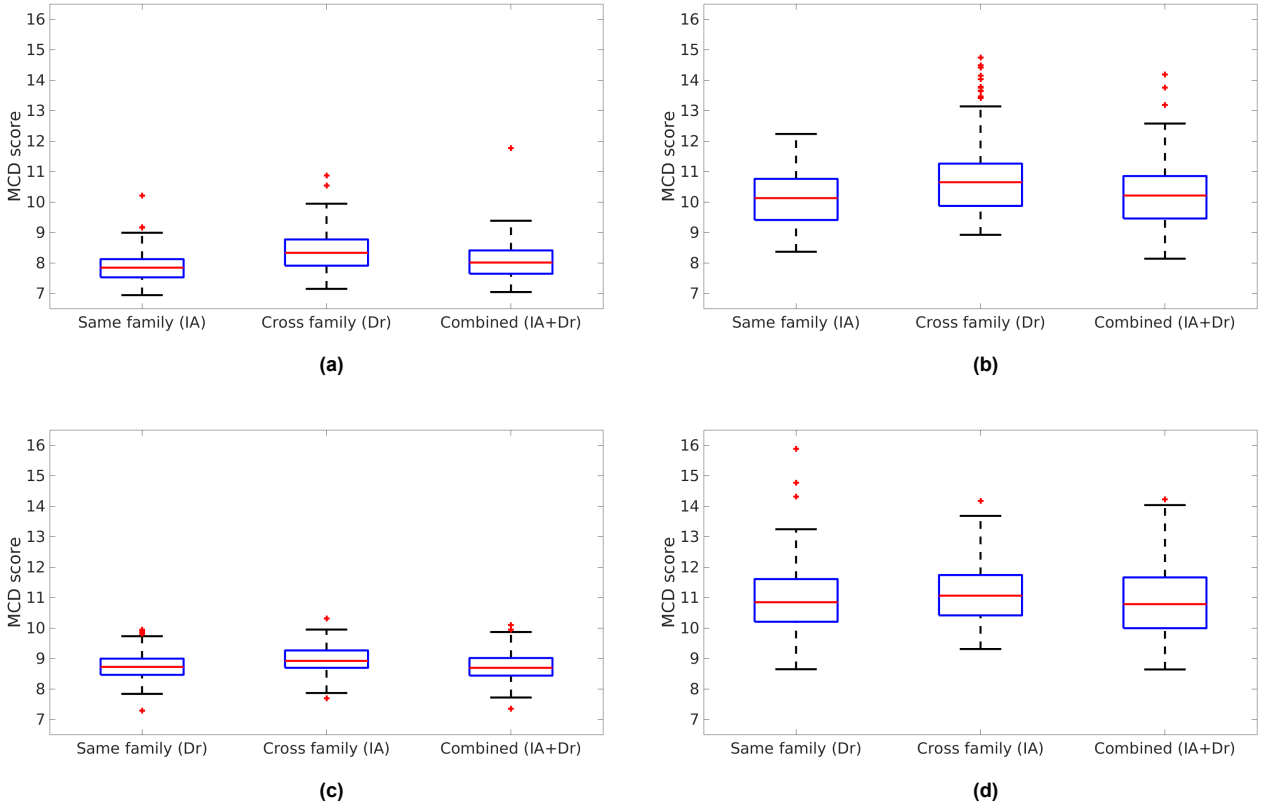

Fig. 6: MCD scores of TTS systems adapted from different generic voices—of the same language family, different language family and IA+Dr combined voice. (a) Gujarati (male) (b) Gujarati (female) (c) Tamil (male) (d) Tamil (female).

material). We see that the better the generic models, the better the performance of adapted models.

MUSHRA test: A MUSHRA test is conducted to evaluate the various adapted systems. 8 native Gujarati and 13 native Tamil speakers participated in the study. Each listener assessed a set of 21 audio files (7 each from each system). Results of the MUSHRA test are presented in Figure 7. It is seen that the synthesis quality of cross family adaptation is poor compared to that of the other two adapted voices. The quality of same family adaptation and combined IA+Dr adaptation is similar in most cases. For Tamil female, Dravidian voice adaptation is better than combined IA+Dr adaptation. On closer inspection, we observe that this lower rating for IA+Dr adaptation is mainly due to unnatural pauses in the synthesised audio, which does not show up in the MCD scores (Figure 6 (d)). This needs further investigation.

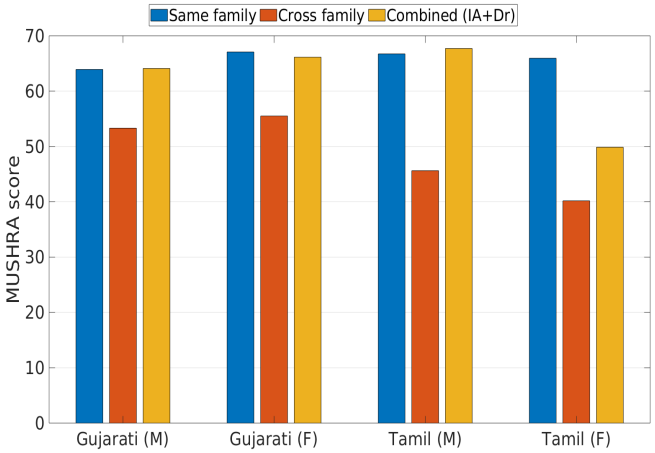

Fig. 7: MUSHRA scores of monolingual text of Gujarati and Tamil synthesised by various adapted systems—same family, cross family and combined IA+Dr voice adaptations (M- male, F- female)

VI. Discussion

In this work, we train and analyse TTS systems for Indian languages in a low-resource setting from a language family perspective. Our observations based on this work are summarised here:

- Monolingual systems built using 5 hours of studio-recorded data and accurate transcriptions produce better quality speech than single-family voices in most cases. Including data from other languages for training could introduce ambiguity. However, the degradation in the quality of single-family systems is marginal. Both monolingual and single-family systems are on par in terms of synthesis quality.
- Single-family voices work reasonably well for unseen languages belonging to the same language family.
- Cross-language family synthesis performs poorly. A contributing factor is a mismatch between the phone-sets of a single-family voice and an unseen language of another language family. The phoneset of an unseen language is largely covered by other languages in the same language family. The phoneset of Gujarati is covered in the Indo-Aryan voice, while Tamil has additional phones such as “e” (short E), “o” (short O) and “zh” (retroflex continuant), which are not covered. Table VII provides details of phone coverage for each unseen language with respect to single-family data. Accent is also an important factor that contributes to poor intelligibility.

TABLE VII: Percentage of unique phones covered by single-family voices. The number of phones that are not covered is given in parentheses.

Unseen language	IA data	Dr data
Gujarati	100% (0)	94% (3)
Tamil	90.48% (4)	100% (0)

- Single-family voices of reasonable quality can be trained in data-stressed situations. With an average duration of 1.5 hours per language (33.33% of monolingual data) for training, the MCD score of the single-family (5hrs) voice has an average relative degradation of only 4.16% in comparison to a monolingual voice built with the same amount of collective data (i.e., 5 hours). The synthesis quality of these systems has an average relative degradation of 12.93% compared to the best TTS system.
- Combined IA+Dr voices, which include languages from both Indo-Aryan and Dravidian language families, do not give a significant performance boost compared to systems trained solely from the same language families.
- Generic voices can be adapted to new languages with limited data. Adaptation is effective when the generic voice is trained on languages similar to the new language. Language families play a vital role here.
- In Tamil, the same character represents both voiced and unvoiced stop consonants. For example, the bilabial unvoiced stop consonant “p” and its voiced counterpart “b” are represented by a single character. This distinction is made in the phone-based representation. Hence, Dravidian (CLS) voice is better suited for adaptation to Tamil.
- In Malayalam and Tamil, when “u” occurs at the end of a word, it is not rounded but uttered as an unrounded back vowel. In most cases of Tamil synthesis using Dr and IA+Dr voices, the vowel “u” remains rounded. With more adaptation data, the network synthesises these Tamil words correctly.
- Generic and adapted systems trained with x-vectors as speaker embedding have lower MCD scores than counterparts without speaker embedding (Sections S1, S2 and S4 of supplementary material). Nonetheless, language family-based analysis still holds for voices with x-vectors.
- Training a single female+male voice with speaker embedding also does not seem to improve the performance of generic systems (Sections S1 and S2 of supplementary material).

A. Analysis of phonotactics across languages

To better understand the outcome of the experiments described above from a theoretical perspective, we analyse the phonotactics of languages and compare them in a multilingual setting. As mentioned in Section II, phonotactics play a vital role in a language. Here we quantify phonotactics using two approaches—byte-pair encoding (BPE) [11] and phone-based language modelling. This is a text-based analysis. The text is first parsed into its phone-based representation. Along with the data used earlier for training and testing, additional text material is used for this analysis—(a) 150 test sentences in Bengali and Telugu (b) training text corresponding to a 5-hour duration in Gujarati and Tamil. Corresponding language data are combined for Indo-Aryan (IA), Dravidian (Dr) and IA+Dr text. Multilingual text data exclude Gujarati and Tamil, which are still considered unseen languages.

1) Analysis of byte-pair encoding (BPE): BPE is a technique originally used for data compression [10], and now adopted for subword tokenization in machine translation [11] and speech-related tasks [69]. BPE tokens can represent the most common sub-strings of a language. These tokens are extracted for every language using their corresponding training text. We consider the top 500 BPE tokens. It is to be noted that this analysis does not include any test data and is performed only on the training text.

TABLE VIII: Same BPE tokens across pairs of text data (values in %)

Language	IA	Dr	IA+Dr
Bengali	**61.0**	36.0	51.8
Hindi	**61.6**	31.4	52.2
Odia	**54.6**	31.2	45.2
Rajasthani	**61.2**	34.2	55.0
Gujarati (unseen)	**52.4**	36.6	49.8
Kannada	36.0	**57.8**	48.2
Malayalam	31.6	**48.8**	42.0
Telugu	32.2	**54.4**	45.2
Tamil (unseen)	28.8	33.6	**35.4**

The percentage of the same BPE tokens is calculated for every combination pair of IA/Dr/IA+Dr data and individual language data. Table VIII presents the results of BPE analysis. We see a higher match for languages to their corresponding language family data compared to the other language family data. For IA+Dr data, this percentage is between individual IA and Dr data. Similar results are also observed for Gujarati, which is not seen in the IA text. The only exception is Tamil, in which the matching percentage is slightly improved for IA+Dr data compared to individual Dr data.

2) Analysis of language models: A phone-level language model (LM) is trained using the corresponding training text for each multilingual data (IA/Dr/IA+Dr). Average sentence-level log-likelihood scores are calculated on the test data using these models. Language models are also trained on monolingual text to understand the maximum achievable likelihood scores. The models are trained and tested using the Stanford Research Institute language modelling (SRILM) toolkit [70] with the maximum order being 3 (i.e., up to trigrams).

TABLE IX: Log likelihood scores of test sets using different language models (maximum order of n-gram= 3).

Test set/ LM	Monolingual	IA	Dr	IA+Dr
Bengali	-65.40	**-71.31**	-112.61	-74.47
Hindi	-90.63	**-97.54**	-159.75	-102.25
Odia	-66.76	**-70.80**	-114.86	-74.67
Rajasthani	-103.40	**-107.00**	-161.47	-111.19
Gujarati (unseen)	-72.60	**-90.09**	-118.17	-90.87
Kannada	-64.62	-124.19	**-71.03**	-76.79
Malayalam	-58.99	-120.68	**-64.93**	-69.79
Telugu	-111.22	-170.39	**-114.68**	-120.56
Tamil (unseen)	-93.25	-171.02	-135.23	**-125.57**

Table IX presents the log-likelihood scores of different test sets using various models. As expected, the likelihood scores of purely monolingual models are the best. IA models have better scores for Indo-Aryan (seen) languages than Dr models. IA+Dr model has slightly lower likelihood scores compared to the IA language model. A similar trend is observed for Dravidian (seen) languages. Even for Gujarati, whose text is not seen in any multilingual language model, the IA model has the best score among the multilingual models. For Tamil, the best multilingual model is IA+Dr. Also, the difference between the likelihood values of the IA+Dr/Dr model and the monolingual Tamil model is relatively high, even for unseen languages. This indicates that the phonotactics of Tamil could perhaps be quite different compared to other Dravidian languages. Overall, the above phonotactic analysis provides a basis for multilingual system training based on language families.

This work shows that language families are important for system building, especially in resource-scarce scenarios. A suitable starting point to build a TTS synthesiser for a new language with limited data would be to use a generic voice trained for the same language family. This would ensure that similar phonotactics are largely covered (Sections II and VI-A), with the added advantage of reducing the overall training data requirement.

Going ahead, the training data per language can be further reduced to assess extreme data-stressed situations. To improve the synthesis quality of seen languages, generic voices can be further fine-tuned on seen languages, as explored in [28], [56]. Additional embeddings, such as language embeddings, can be included during training. The code-mixing ability of generic voices can also be explored. Given data in more Indian languages, the study can be extended to include more language combinations. Even with recent approaches using transformer [71] and conformer [72] networks with FastSpeech [73] and FastSpeech2 [74], these findings are still relevant. Experiments and results of zero-shot synthesis with transformer-based FastSpeech2 architecture are presented in Section S7 of the supplementary material. Since FastSpeech2 uses explicit phoneme boundaries obtained from Montreal forced aligner [75], systems are trained using phone-based representations.

VII. Conclusion

This work highlights the importance of training multilingual and multispeaker voices for low-resource Indian languages based on language families. It is observed that

single-family voices, which are trained on less data, perform comparatively to IA+Dr systems trained on a lot of data. Same language family synthesis and adaptation are better than the cross-family approach. The observations of this work are encouraging as they pave the way to training TTS systems in resource-scarce scenarios, with additional complexities of different scripts and language-specific differences.

Given a large number of speakers in each language, with many that cannot read or write in India[5], this work provides an avenue to disseminate knowledge and information. Hence, the relevance of this work and similar attempts cannot be underestimated.

Acknowledgment

This work was undertaken as part of the following projects funded by different agencies: (1) "Text to Speech Generation with chosen accent and noise profile for Aerospace and Industrial domains" (IMP/2018/000986) by Department of Science and Technology (DST), Government of India (GoI), (2) "Natural Language Translation Mission" (11(1)/2020-HCC(TDIL)) by the Ministry of Electronics and Information Technology (MeitY), GoI, (3) "Speech to Speech Machine Translation" (SA/NL/2018(G)&(C)) by Office of the Principal Scientific Adviser (PSA), GoI, and (4) "Speech Technologies in Indian Languages" (SP/21-22/1960/CSMEIT/003119) by MeitY, GoI.

References

[1] Y. Wang et al., "Tacotron: Towards end-to-end speech synthesis," in *INTERSPEECH*, 2017, pp. 4006–4010.

[2] J. Shen *et al.*, "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] Government of India, "Census of India 2011," https://censusindia.gov.in/2011Census/Language_MTs.html, 2011.

[4] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. Skerry-Ryan, "Semi-Supervised Training for Improving Data Efficiency in End-to-End Speech Synthesis," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6940–6944.

[5] A. Prakash and H. A. Murthy, "Generic Indic Text-to-Speech Synthesisers with Rapid Adaptation in an End-to-End Framework," in *INTERSPEECH*, 2020, pp. 2962–2966.

[6] A. Prakash, A. Leela Thomas, S. Umesh, and H. A Murthy, "Building Multilingual End-to-End Speech Synthesisers for Indian Languages," in *10th ISCA Speech Synthesis Workshop (SSW)*, 2019, pp. 194–199.

[7] B. Ramani *et al.*, "A common attribute based unified HTS framework for speech synthesis in Indian languages," in *Speech Synthesis Workshop (SSW)*, 2013, pp. 291–296.

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[9] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *INTERSPEECH*, 2017, pp. 999–1003.

[10] P. Gage, "A New Algorithm for Data Compression," *C Users J.*, vol. 12, no. 2, p. 23–38, 1994.

[11] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2016, pp. 1715–1725.

[5]Literacy rate in India is 74.04% [3]

[12] A. Prakash, J. J. Prakash, and H. A. Murthy, "Acoustic Analysis of Syllables Across Indian Languages," in *INTERSPEECH*, 2016, pp. 327–331.

[13] A. Sen and K. Samudravijaya, "Indian accent text-to-speech system for web browsing," *Sadhana*, vol. 27, no. 1, pp. 113–126, 2002.

[14] S. Davis, *Geminates*. American Cancer Society, 2011.

[15] M. Ohala, *Aspects of Hindi phonology*. Motilal Banarsidass Publishers Pvt. Ltd, 1983.

[16] B. Krishnamurti, *The Dravidian languages*. Cambridge University Press, 2003.

[17] C. Traber, K. Huber, K. Nedir, B. Pfister, E. Keller, and B. Zellner, "From multilingual to polyglot speech synthesis," in *European Conference on Speech Communication and Technology*, 1999.

[18] J. Latorre, K. Iwano, and S. Furui, "Cross-language synthesis with a polyglot synthesizer," in *INTERSPEECH*, 2005, pp. 1477–1480.

[19] J. Latorre, K. Iwano, and S. Furui, "Polyglot synthesis using a mixture of monolingual corpora," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005, pp. 1–4.

[20] J. Latorre, I. Koji, and S. Furui, "New approach to the polyglot speech generation by means of an HMM-based speaker adaptable synthesizer," *Speech Communication*, vol. 48, no. 10, pp. 1227–1242, 2006.

[21] H. Zen, N. Braunschweiler, S. Buchholz, M. J. F. Gales, K. Knill, S. Krstulovic, and J. Latorre, "Statistical Parametric Speech Synthesis Based on Speaker and Language Factorization," *IEEE Trans. Audio, Speech & Language Processing*, vol. 20, no. 6, pp. 1713–1724, 2012.

[22] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Speaker and language factorization in DNN-based TTS synthesis," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5540–5544.

[23] P. Vijayalakshmi, B. Ramani, M. A. Jeeva, and T. Nagarajan, "A multilingual to polyglot speech synthesizer for Indian languages using a voice-converted polyglot speech corpus," *Circuits, Systems, and Signal Processing*, vol. 37, no. 5, pp. 2142–2163, 2018.

[24] E. Nachmani and L. Wolf, "Unsupervised Polyglot Text To Speech," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7055–7059.

[25] J. Yang and L. He, "Towards Universal Text-to-Speech," in *INTERSPEECH*, 2020, pp. 3171–3175.

[26] T. Nekvinda and O. Dušek, "One Model, Many Languages: Meta-learning for Multilingual Text-to-Speech," in *INTERSPEECH*, 2020, pp. 2972–2976.

[27] J. Fong, J. Wu, P. Agrawal, A. Gibiansky, T. Koehler, and Q. He, "Improving Polyglot Speech Synthesis through Multitask and Adversarial Learning," in *Speech Synthesis Workshop (SSW)*, 2021, pp. 172–176.

[28] J. Latorre, C. Bailleul, T. Morrill, A. Conkie, and Y. Stylianou, "Combining speakers of multiple languages to improve quality of neural voices," in *Proc. 11th ISCA Speech Synthesis Workshop (SSW)*, 2021, pp. 37–42.

[29] M. Chu, H. Peng, Y. Zhao, Z. Niu, and E. Chang, "Microsoft Mulan-a bilingual TTS system," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003, pp. 264–267.

[30] H. Liang, Y. Qian, and F. K. Soong, "An HMM-based Bilingual (Mandarin-English) TTS," in *Speech Synthesis Workshop (SSW)*, 2007, pp. 137–142.

[31] A. L. Thomas, A. Prakash, A. Baby, and H. A. Murthy, "Code-switching in Indic Speech Synthesisers," in *INTERSPEECH*, 2018, pp. 1948–1952.

[32] Y. Cao et al., "End-to-end Code-switched TTS with Mix of Monolingual Recordings," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6935–6939.

[33] L. Xue, W. Song, G. Xu, L. Xie, and Z. Wu, "Building a mixed-lingual neural TTS system with only monolingual data," in *INTERSPEECH*, 2019, pp. 2060–2064.

[34] S. Bansal, A. Mukherjee, S. Satpal, and R. Mehta, "On improving code mixed speech synthesis with mixlingual grapheme-to-phoneme model," in *INTERSPEECH*, 2020, pp. 2957–2961.

[35] M. Staib, T. H. Teh, A. Torresquintero, D. S. R. Mohan, L. Foglianti, R. Lenain, and J. Gao, "Phonological features for 0-shot multilingual speech synthesis," in *INTERSPEECH*, 2020, pp. 2942–2946.

[36] S. Zhao, T. H. Nguyen, H. Wang, and B. Ma, "Towards natural bilingual and code-switched speech synthesis based on mix of monolingual recordings and cross-lingual voice conversion," in *INTERSPEECH*, 2020, pp. 2927–2931.

[37] S. S. Sarfjoo and C. Demiroglu, "Cross-Lingual Speaker Adaptation for Statistical Speech Synthesis Using Limited Data," in *INTERSPEECH*, 2016, pp. 317–321.

[38] Y. Zhao, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Kinnunen, Z. Ling, and T. Toda, "Voice conversion challenge 2020: Intra-lingual semi-parallel and cross-lingual voice conversion," in *Voice Conversion Challenge*, 2020.

[39] E. V. Raghavendra et al., "Global syllable set for building speech synthesis in Indian languages," in *IEEE Spoken Language Technology Workshop*, 2008, pp. 49–52.

[40] Q. Yu, P. Liu, Z. Wu, S. K. Ang, H. Meng, and L. Cai, "Learning cross-lingual information with multilingual BLSTM for speech synthesis of low-resource languages," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5545–5549.

[41] B. Li and H. Zen, "Multi-Language Multi-Speaker Acoustic Modeling for LSTM-RNN Based Statistical Parametric Speech Synthesis," in *INTERSPEECH*, 2016, pp. 2468–2472.

[42] A. Gutkin, "Uniform Multilingual Multi-Speaker Acoustic Model for Statistical Parametric Speech Synthesis of Low-Resourced Languages," in *INTERSPEECH*, 2017, pp. 2183–2187.

[43] I. Demirsahin, M. Jansche, and A. Gutkin, "A Unified Phonological Representation of South Asian Languages for Multilingual Text-to-Speech," in *Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, 2018, pp. 80–84.

[44] M. de Korte, J. Kim, and E. Klabbers, "Efficient Neural Speech Synthesis for Low-Resource Languages Through Multilingual Modeling," in *INTERSPEECH*, 2020, pp. 2967–2971.

[45] H. Ming, Y. Lu, Z. Zhang, and M. Dong, "A light-weight method of building an LSTM-RNN-based bilingual TTS system," in *International Conference on Asian Language Processing (IALP)*, 2017, pp. 201–205.

[46] I. Himawan, S. Aryal, I. Ouyang, S. Kang, P. Lanchantin, and S. King, "Speaker Adaptation of a Multilingual Acoustic Model for Cross-Language Synthesis," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7629–7633.

[47] S. Maiti, E. Marchi, and A. Conkie, "Generating multilingual voices using speaker space translation based on bilingual speaker data," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7624–7628.

[48] J. Williams, J. Fong, E. Cooper, and J. Yamagishi, "Exploring Disentanglement with Multilingual and Monolingual VQ-VAE," in *Proc. 11th ISCA Speech Synthesis Workshop (SSW)*, 2021, pp. 124–129.

[49] B. Li, Y. Zhang, T. Sainath, Y. Wu, and W. Chan, "Bytes are All You Need: End-to-End Multilingual Speech Recognition and Synthesis with Bytes," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5621–5625.

[50] G. Maniati *et al.*, "Cross-Lingual Low Resource Speaker Adaptation Using Phonological Features," in *INTERSPEECH*, 2021, pp. 1594–1598.

[51] Y. Z. et al., "Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning," in *INTERSPEECH*, 2019, pp. 2080–2084.

[52] D. Xin, Y. Saito, S. Takamichi, T. Koriyama, and H. Saruwatari, "Cross-Lingual Text-To-Speech Synthesis via Domain Adaptation and Perceptual Similarity Regression in Speaker Space," in *INTERSPEECH*, 2020, pp. 2947–2951.

[53] D. Xin et al., "Disentangled Speaker and Language Representations Using Mutual Information Minimization and Domain Adaptation for Cross-Lingual TTS," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6608–6612.

[54] P. Baljekar, S. K. Rallabandi, and A. W. Black, "An Investigation of Convolution Attention Based Models for Multilingual Speech Synthesis of Indian Languages," in *INTERSPEECH*, 2018, pp. 2474–2478.

[55] M. Chen, M. Chen, S. Liang, J. Ma, L. Chen, S. Wang, and J. Xiao, "Cross-Lingual, Multi-Speaker Text-To-Speech Synthesis Using Neural Speaker Embedding," in *INTERSPEECH*, 2019, pp. 2105–2109.

[56] K. Prajwal and C. Jawahar, "Data-Efficient Training Strategies for Neural TTS Systems," in *8th ACM IKDD CODS and 26th COMAD*, 2021, pp. 223–227.

[57] D. Wells and K. Richmond, "Cross-lingual Transfer of Phonological Features for Low-resource Speech Synthesis," in *Proc. 11th ISCA Speech Synthesis Workshop (SSW)*, 2021, pp. 160–165.

[58] M. He, J. Yang, and L. He, "Multilingual byte2speech text-to-speech models are few-shot spoken language learners," 2021. [Online]. Available: https://arxiv.org/abs/2103.03541

[59] Z. Shang, Z. Huang, H. Zhang, P. Zhang, and Y. Yan, "Incorporating Cross-Speaker Style Transfer for Multi-Language Text-to-Speech," in *INTERSPEECH*, 2021, pp. 1619–1623.

[60] P. Do, M. Coler, J. Dijkstra, and E. Klabbers, "A Systematic Review and Analysis of Multilingual Data Strategies in Text-to-Speech for Low-Resource Languages," in *INTERSPEECH*, 2021, pp. 16–20.

[61] Z. Zhang, A. Falai, A. Sanchez, O. Angelini, and K. Yanagisawa, "Mix and Match: An Empirical Study on Training Corpus Composition for Polyglot Text-To-Speech (TTS)," in *INTERSPEECH*, 2022, pp. 2353–2357.

[62] 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.

[63] A. Baby, N. N. L., A. L. Thomas, and H. A. Murthy, "A unified parser for developing Indian language text to speech synthesizers," in *International Conference on Text, Speech and Dialogue*, 2016, pp. 514–521.

[64] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A Flow-based Generative Network for Speech Synthesis," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3617–3621.

[65] T. Hayashi et al., "Espnet-TTS: Unified, Reproducible, and Integratable Open Source End-to-End Text-to-Speech Toolkit," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7654–7658.

[66] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[67] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, 1993, pp. 125–128.

[68] ITU-R Recommendation, "BS. 1534-1. Method for the subjective assessment of intermediate sound quality (MUSHRA)," in *International Telecommunications Union*, 2001.

[69] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved Training of End-to-end Attention Models for Speech Recognition," in *INTERSPEECH*, 2018, pp. 7–11.

[70] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *International Conference on Spoken Language Processing (ICSLP)*, 2002, pp. 901–904.

[71] A. Vaswani et al., "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[72] A. Gulati et al., "Conformer: Convolution-augmented Transformer for Speech Recognition," in *INTERSPEECH*, 2020, pp. 5036–5040.

[73] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "FastSpeech: Fast, Robust and Controllable Text to Speech," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019, pp. 3165–3174.

[74] Y. R. et al., "FastSpeech 2: Fast and High-Quality End-to-End Text to Speech," in *International Conference on Learning Representations, (ICLR)*, 2021.

[75] M. McAuliffe et al., "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi," in *INTERSPEECH*, 2017, pp. 498–502.