INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

On Improving Code Mixed Speech Synthesis with Mixlingual Grapheme-to-Phoneme Model

Shubham Bansal†*, Arijit Mukherjee*†*, Sandeepkumar Satpal, Rupeshkumar Mehta*

Microsoft STC India

{shbansa, armuk, ssatpal, rupeshme}@microsoft.com

Abstract

Regional entities often occur in a code-mixed text in the non-native roman script and synthesizing them with the correct pronunciation and accent is a challenging problem. English grapheme-to-phoneme (G2P) rules fail for such entities because of the orthographical mistakes and phonological differences between the English and regional languages. The traditional approach for this problem involves language identification, followed by the transliteration of the regional entities to their native language and then passing them through a native G2P. In this work, we simplify this module based architecture by learning an end-to-end mixlingual G2P in a multi-task type setting. Also, rather than mapping the output phone sequences from our mixlingual G2P to the English phoneset or using the "shared" phoneset, we use the polyglot data and "separated" phoneset to train a mixlingual synthesizer to improvise the synthesized voice accent for regional entities. We have used Hindi-English as the code-mix scenario and we show absolute incremental gains of up to 28% in pronunciation accuracy and a 0.9 gain in "overall impression" mean-opinion-score (MOS) over using a standard English monolingual text-to-speech (TTS).

Index Terms: text-to-speech, mixlingual, regional, code-mixing, grapheme-to-phoneme

1. Introduction

Mixing the words from the regional languages in the English language while speaking or writing is an example of a code-mixing and is a very common phenomenon in a conversational speech or text. We call such words as "regional entities". For instance, a following Facebook post from a person with 29 million followers consists of both Hindi(originally written in the "Devanagari" script) and English words written in the "roman" script:*"The front line workers.. the nurses and doctors.. the social warriors. natmastak hoon main"*. Here, *"natmastak hoon main"* is the regional entity and it translates to *"I bow before you"*. Grapheme-to-phoneme (G2P) models convert a written word into its corresponding pronunciation and are essential component in the text-to-speech (TTS) systems. English G2P model trained only on the English entities and also using the combined phoneset [1, 2] to handle the phonological differences does not perform well on the regional entities written in roman script broadly because of the three reasons: (1) Correct pronunciation for the regional entities is often present in the broader phonological inventory and several non-English phones are rarely seen in the G2P training despite using the combined phoneset. (2) Orthographical mistakes. For example, alphabet *"n"* in the word *"hoon"* represents a nasalized vowel and such use is not standardized by the English orthography rules and (3) Non-phonetic nature of the English language [3] which further makes the G2P model learning more difficult. In this work, we have used "Hindi" as the regional language and "Hindi-English" (Hinglish) as the code-mix scenario.

Pronunciation for regional entities written in roman script are more accurately obtained by identifying [4, 5, 6] them from a given code-mixed text, transliterating [7, 8] to their native language and finally using the native G2P to obtain the pronunciation in the phonology of the regional language. This modular approach [2, 9, 10, 11, 12] is carried out in a sequential way which results in increase in the latency and a high cumulative error on the regional entity pronunciation task. Also, each of the module is optimized separately with its own objective, which may result in incoherence in optimization, as no module is trained to match the other modules [13]. In this work, we propose a technique to leverage the modular approach [2, 11] to efficiently generate the pronunciation for several such regional words written in roman script for training an end-to-end mixlingual G2P and show that it improves the language identification and pronunciation accuracy significantly for both the English and regional entities in a code-mixed text. Another challenge in the code-mixed speech synthesis is to have a synthesizer that can handle the phonology of both the regional language and English language. Earliest approaches maps the output phones from the native G2P to the most similar sounding phones in the English language [14, 15, 16, 9] and train a monolingual synthesizer [9] resulting in an undesirable foreign accent for the regional entities. Other approaches involves using the combined phoneset [2, 11] and training a mixlingual synthesizer either with the synthesized polyglot data [10] or using the separate recordings [2, 11] for both the languages in the voice of a single speaker. In the combined phoneset , common sounds across both the languages are mapped together with the same phones [1] and the disjoint phones are added separately. Combined phoneset at this stage is also called "shared" phoneset. Further, when we add the language tags: "en_" and "hi_" for every phone in the "shared" set, it is called the "separated" phoneset. Previous works [2, 11] suggests using the "shared" phoneset on the basis of a preference test. In this work, we propose using the "separated" phoneset on the basis of a mean opinion score (MOS) study on 3 different test sets. We train a Transformer TTS [17] based mixlingual synthesizer using a polyglot data similar to [2] and show that we synthesize the code-mixed speech very similar in naturalness to the human recordings. Cross-lingual models [18] may also be explored but they need multiple speakers from both the languages to train and also have been shown to achieve an average voice similarity score during the cross-lingual voice transfer which is not very desirable for code-mixed speech synthesis and are most ideal in the scenario when it is hard to obtain the polyglot data.

The rest of paper is organized as follows. In Sec. 2, we brief about the the neural TTS system. Sec. 3 describes the techniques to synthesize the training data and training of mixlingual

†: Both authors have contributed equally.

http://dx.doi.org/10.21437/Interspeech.2020-2654

G2P model. We report the experiments and results in Sec. 4. In Sec. 5, we conclude with the summary of the work.

2. Neural Text-to-Speech

Neural text-to-speech (TTS) uses deep neural networks to synthesize the audio waveform from a given text. Most of the reliable models works on the modular approach. It includes two main components, (1) Front-end: a)Text normalizer and b)G2P, (2) Synthesizer : a) Acoustic Model [19, 20, 21, 17, 22] and b) Vocoder [23, 24, 25].

2.1. Monolingual English Neural TTS

Techniques used to improve the pronunciation and accent for the regional entities in a code-mixed scenario may have an adverse affect on the synthesis quality of the English entities. To set our benchmark for the English speech synthesis, we build a monolingual English TTS system. Our monolingual English front-end comprises a text normalizer, an English lexicon and a Transformer-based G2P model [26] trained with the word-pronunciation pairs present in an English lexicon. We use the combined phone-set [1, 2] and also ensure that the words used to train the English G2P strictly follows the English orthography rules. Further, we train an auto regressive Transformer based acoustic model[17] with 5000 utterances from an internal female English speaker. We also train a WaveNet Vocoder [23] conditioned on Mel spectrograms with similar parameters from Li et al. [17] with the same 5000 utterances.

3. Systems Overview

In this section, we describe the architecture of the previous and our proposed approach to handle the regional entities in a code-mixed text.

3.1. Identify-Transliterate-Phonate (ITP) Front-End

Previous works [2, 11, 12] suggested using a language identification model to identify the regional entities in a code-mixed text. Once identified, English entities are passed through an English G2P and those identified as the regional entities are transliterated into their native script and further passed through a native G2P. We henceforth call this approach as identify-transliterate-phonate (ITP) model for our convenience. This approach is very intuitive but its efficacy is strictly dependent on the accuracy of each of its components. We discuss the details of each of the modules in the following subsections.

3.1.1. Language Identification model (LID)

Language identification is a task of identifying the language of origin of a given word. We use the LID model by Gella et al. [27] which was the best performing system in the FIRE 2013 Hindi-English language identification task [28]. The system uses a maximum entropy classifier trained on the character level n-grams from Hindi and English words. In our experiments, we also use a dictionary of approximately 170k English words to further improve the accuracy of the LID model.

3.1.2. Transliteration model

Previous related works [11] explored several transliteration schemes and suggested using the Brahmi-net transliteration [7] which considers the problem of transliteration similar to a phrase based translation problem and reported the word accuracy of 32.65% for an English-to-Hindi transliteration task. Even the attention-RNN network based transliteration systems [8] and models [29] from the recent "named entity transliteration" shared task [30] reports less than 60% transliteration accuracy for several language pairs. Additionally, there is a need of significant amount of linguistic resources in the form of parallel training data to train a transliteration model. In this work, we use the latest English-to-Hindi transliteration API deployed on Microsoft azure cognitive services for our experiments.

3.1.3. Native G2P

We train a transformer based G2P similar to Sun et al. [26] using the word-pronunciation pairs from a Hindi lexicon. Also, most Indic languages are very phonetic which results in a very high accuracy for the Indic G2P models. We achieve 98.3% word level accuracy with our Hindi G2P model on an internal test set. Also, we use the combined "shared" phoneset as described in Sec. 1 inline with the previous works [2, 11] to train a Hindi G2P model to handle the phonological differences with its English counterpart.

From the architecture, we may conclude that the pronunciation accuracy for the regional entities in a code-mixed text is restricted by the accuracy of each of the modules. Further, using a modular approach may also result in the increase in latency for an overall TTS system. Our ITP front-end also uses an English lexicon and a text normalizer in addition to the above modules.

3.2. Proposed Mixlingual Front-End

We leverage an ITP model to generate the word-pronunciation pairs for training a single transformer-based network "mixlingual G2P" similar to Sun et al. [26] which jointly learns pronunciation and language identification. We also use the combined phoneset as described in Sec. 1.

3.2.1. Training data synthesis

Training such a network requires a large number of word-pronunciation pairs of the English and regional entities. Such pairs for English entities are easily obtained from an English lexicon whereas it is hard to obtain similar data for the regional entities written in the roman script as each pronunciation has to be hand-crafted by a language expert with the knowledge of phonology and orthography of both the languages. In this work, we leverage an ITP model and exploit the high accuracy of the native G2P to generate very accurate regional word (roman script)-pronunciation pairs as shown in Fig. 1. We use an English language modelling corpus that has a good distribution of the regional entities and sort the uni-grams in the order of their frequency before passing them through a LID followed by a transliteration module. Evaluating the transliteration and language of origin does not require any language phonetics expertise and is easily performed through crowd-sourcing from the native speakers. We further pass correctly identified transliterated words to a native G2P and also add a language tag "hi_" for each phone in the pronunciation to incorporate the language information which in our case is "Hindi" as shown in Fig. 1. For English entities, we use an English lexicon and add a language tag "en_". We also compare the efficiency and accuracy of this approach against the hand-crafted approach in generating the pronunciations of randomly selected 100 regional entities written in roman script in Table 1.

3.2.2. Mixlingual G2P Training

We train a transformer model similar to Sun et al. [26]. In addition to that work, we have a language tag in the pronunciation as explained in the previous section that enables our network to learn the language dependent character-sequence to phone-

Randomly Selected 100 Reg. Entities	Proposed	Hand-crafted
Time Taken(in Minutes)	18 min	92 min
Word-level Pronunciation Accuracy	96%	100%

Table 1: *Accuracy and efficiency comparison between ITP (transliteration and language tag evaluated) and hand-crafted approach.*

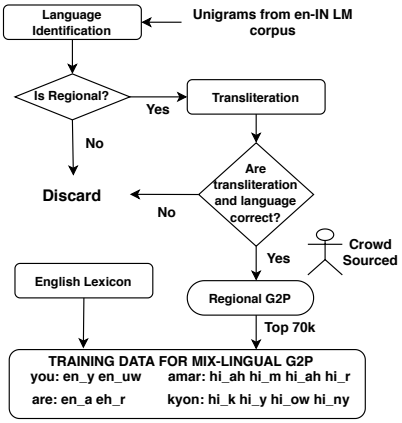

Figure 1: *Leveraging ITP model for efficiently generating highly accurate regional word-pronunciation pairs*

sequence mappings. Also, our validation set consists only of English word-pronunciation pairs to minimize the regression on the pronunciation accuracy of English entities because of the combined training.

3.2.3. Inference

Our mixlingual front-end comprises of a text normalizer, an English lexicon and a mixlingual G2P module. During inference, if a word is present in an English lexicon, we output a language tag "en" and pronunciation as defined in the lexicon and if not, we pass it through our mixlingual G2P model. We finally obtain the language tag (English or regional) and the pronunciation (phone sequence) by separating them from the output of the mixlingual G2P as also shown in Fig. 2. For example, a mixlingual model output "hi_ah hi_m hi_ah hi_r" corresponding to a word "Amar" splits into a language tag "hi" and a phone sequence "ah m ah r". Further, depending on whether our synthesizer use the "shared" or "separated" phoneset, we add back the language tag or pass it as such to the synthesizer. The motive behind separating the language tag from the mixlingual G2P output is that many regional entities when written in the roman script such as "मै" written as "main", "है" written as "he" are also valid English entities but are pronounced differently. We refer to this phenomenon as "language-polyphony". We have created a list of such regional entities and we use the language tag information of the neighborhood words to disambiguate between the language tag and pronunciation of such words.

3.3. Mixlingual synthesizer

Our next step is to synthesize a speech from an output phone-sequence. Previous works [2, 10, 11] showed that training the mixlingual synthesizer with a polyglot voice improves the accent of the regional entities in a code-mixed text. Further, Rallabandi et al. [2] suggested using a "shared" phoneset over a "separated" phoneset for both the ITP front-end and the mixlingual synthesizer to handle the phonological differences between the two languages. Our mixlingual front-end outputs a phone-sequence on the "shared" phoneset and also a language tag corresponding to each word. We use the language tag for language-

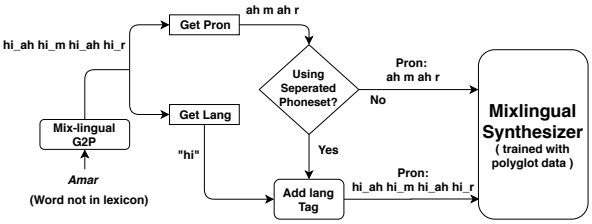

Figure 2: *Combining mixlingual G2P and mixlingual synthesizer*

polyphony disambiguation and also to map the output phone-sequence on the "separated" phoneset if required as shown in Fig. 2. Our mixlingual synthesizer is trained on similar architecture as described in Sec. 2.1. Both the AM and vocoder are trained on the 5000 English + 5000 Hindi text-wave pairs from an internal female speaker. We also experimented with both "shared" and "separated" phoneset in the training of mixlingual synthesizer and we suggest using the "separated" phoneset on the basis of a MOS analysis as discussed in Sec. 4.

4. Experiments and Results

4.1. Pronunciation Evaluation

To evaluate the pronunciation accuracy, we have created the "English" and "Regional" test-sets with 800 and 400 word-pronunciation pairs in them, respectively. For the "Regional" test-set, we have mined the commonly used regional entities written in roman script from different websites. Also, for each word in the "Regional" test-set, we have carefully added all the valid pronunciations. For the "English" test-set, we have mined the valid English words from an English LM. Our baseline is a monolingual English G2P as described in the Sec. 2.1 trained with 130k English word-pronunciation pairs from an English lexicon. For the mixlingual G2P, we train two models one with the additional 20k and other with the additional 70k regional word(roman script)-pronunciation pairs synthesized with the help of an ITP model as described in Sec. 3.2.1. For ITP model, we use the same monolingual English G2P and a native G2P is additionally trained with 70k regional words(native Devanagari script)-pronunciation pairs obtained from a Hindi lexicon. Other modules are used as described in Sec. 3.1. Further, we also report the language identification accuracy and have summarized the results in the Table 2.

	Pron. acc.(%) (English)	LID acc.(%) (English)	Pron. acc.(%) (Regional)	LID acc.(%) (Regional)
Mono G2P(130k Eng.)	77.8	-	41.5	-
Mix G2P(+ 20k Reg.)	77.6	99.97	56.3	89.1
Mix G2P(+ 70k Reg.)	**78.0**	**99.98**	**69.6**	**98.3**
ITP	75.6	89.3	66.6	94.2

Table 2: *Word-level pronunciation accuracy (%) and language identification accuracy (%) comparison among monolingual English G2P, ITP model and mixlingual G2P on the English entities and regional entities testset.*

From the Table 2, we observe 15% and 28% absolute improvement in the pronunciation accuracy of the regional entities written in roman script without significantly regressing on the pronunciation accuracy of the English entities when we add 20k and 70k synthesized regional word-pronunciation pairs in the training of our mixlingual G2P, respectively. We believe that adding more synthesized regional pairs in the mixlingual G2P training will further improve the pronunciation accuracy. Also, we achieve more than 98% accuracy on the LID task with

our mixlingual G2P. Further, our mixlingual G2P is not directly comparable with the described ITP model in Sec. 3.1 because its transliteration and LID modules are pre-trained on an unknown data and its only purpose in this work is to synthesize the regional training data. We observe 25% absolute improvement in the pronunciation accuracy of the regional entities with the described ITP model but also observe a regression of 2% in the pronunciation accuracy of the English entities. This could be attributed to the fact that in ITP model, every miss-classified English entity is transliterated in the script of regional language which could adversely affect its pronunciation accuracy.

4.2. Mean-Opinion-Score (MOS) Evaluation

We also perform a subjective 5-scale MOS test with 14 judges proficient in both the languages and asked to rate the synthesized speech on "overall impression" on 3 different test-sets: "English", "Regional" and "Code-mixed" each comprising of 20 short and long sentences. "English" set comprises of pure English sentences whereas "Regional" set consists of typical Hindi movie dialogues written in roman script. For "Code-mixed" set, we have carefully selected a mix of hard sentences from the FIRE-2014 code-mix shared task [31] and other websites having at least 30% regional entities in each sentence. MOS test is performed on 4 TTS systems: **1) MoFMoS:** Monolingual English neural TTS as described in Sec. 2.1. **2) MiFMoS:** Mixlingual front-end and monolingual English synthesizer **3) MiFMiS_SH:** Mixlingual front-end and mixlingual synthesizer (trained with "shared" phoneset) **4) MiFMiS_SE:** Mixlingual front-end and mixlingual synthesizer (trained with "separated" phoneset). We also compare them with the high-quality recordings of our bilingual voice talent. MOS results are shown in Table 4 and compared in Fig. 3.

Test Set	Sentence
English	How about coming to the barbecue at the club? How bright they've grown in the sunlight!
Code-mixed	tum hamari help karogi toh we all will enjoy. when did sachin hit the last fifty hindustan ke liyen?
Regional (Roman)	Ude Jab Jab Zulfen Teri Rishte mein toh hum tumhare baap lagte hain

Table 3: *Example test sentences(short)*

4.2.1. English Testset

For the "English" test-set, we have used the MOS score of neural monolingual English TTS system (MoFMoS) as our benchmark. We observe that the MOS scores of the MoFMoS and MiFMoS systems are almost similar and we re-emphasize the claim that there is no significant regression in the pronunciation of the English entities generated from a mixlingual G2P. We observe a dip of 0.2 in MOS score of MiFMiS_SH system and on examining the synthesized audios, we observe that training a mixlingual synthesizer with the "shared" phoneset averages out the accent of both the English and the regional language adversely affecting the overall English speech synthesis quality. Further, with a mixlingual synthesizer trained with the "separated" phoneset, we do not observe a significant regression on English accent and is also reflected in the MOS score of MiFMiS_SE system which performs at par with the MoFMoS system for the English sentences.

4.2.2. Code-mixed Testset

In the "Code-mixed" test-set, we observe that the MoFMoS system has the least MOS score and is because of the G2P trained with only English word-pronunciation pairs and has been shown in the Table 2 to have the high pronunciation errors on the regional entities. Hence, incorporating a mixlingual front-end

	English	Code-mixed	Regional
MoFMoS	4.52 ± 0.06	3.18 ± 0.07	2.4 ± 0.08
MiFMoS	4.48 ± 0.05	3.67 ± 0.06	3.26 ± 0.07
MiFMiS_SH	4.33 ± 0.06	3.74 ± 0.07	3.4 ± 0.09
MiFMiS_SE	**4.56 ± 0.07**	**4.05 ± 0.06**	**3.64 ± 0.07**
Human	4.67 ± 0.05	4.62 ± 0.05	4.6 ± 0.04

Table 4: *MOS of TTS systems with 95% confidence interval in our experiments on English, Code-mixed and Regional test-set*

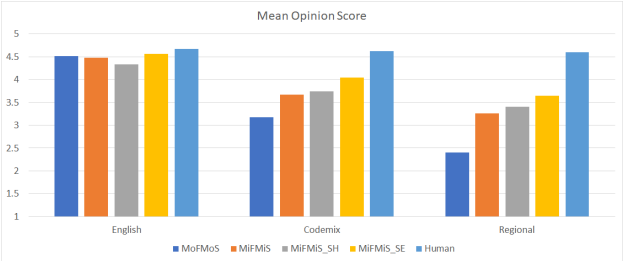

Figure 3: *MOS comparison with different models on the English, Code-mixed and Regional test-set.*

(MiFMoS) gives us 0.45 improvement in MOS score. We observe that the MOS score of MiMiS_SH system is marginally better by 0.08 in comparison to the MiMoS system. We examined the synthesized audio from the MiMiS_SH system and observed that the regional entities still have a significant amount of foreign accent because of the "shared" phoneset. We observe a further improvement of 0.4 of MOS with the MiFMiS_SE system and is attributed to the mixlingual synthesizer trained with the "separated" phoneset which improves the accent of the regional entities without degrading the accent of English entities.

4.2.3. Regional Testset

"Regional" test-set comprises of only regional entities written in the roman script and is the most challenging set for the TTS because of the low G2P pronunciation accuracy and challenges in handling the accent. We observe that our proposed MiFMiS_SE system has the highest MOS score and 1.2 more than the MoFMoS system.

4.2.4. Comparison with Human Recordings

We observe that the proposed mixlingual front-end + mixlingual synthesizer trained with the "separated" phoneset (MiFMiS_SE) system is able to reduce the gap between the human recordings and synthesized speech in both the "Code-mixed" and "Regional" test-set while performing almost at par with the human recordings on a "English" test-set.

5. Conclusion

We propose an architecture, consisting of two components: 1) A mixlingual G2P which learns the pronunciation as well as the language identification in a multitask setup and 2) A mixlingual synthesizer trained with the "separated" phoneset and show that it improves the pronunciation as well as accent of regional entities in a code-mixed sentence significantly.

6. References

[1] B. Ramani, S. L. Christina, G. A. Rachel, V. S. Solomi, M. K. Nandwana, A. Prakash, S. A. Shanmugam, R. Krishnan, S. K. Prahalad, K. Samudravijaya *et al.*, "A common attribute based unified HTS framework for speech synthesis in Indian languages," in *Eighth ISCA Workshop on Speech Synthesis*, 2013.

[2] S. K. Rallabandi and A. W. Black, "On Building Mixed Lingual Speech Synthesis Systems." in *INTERSPEECH*, 2017, pp. 52–56.

[3] R. SUZANA, "Neutralization of mother tongue influence-its importance."

[4] S. Gella, K. Bali, and M. Choudhury, ""ye word kis lang ka hai bhai?" Testing the Limits of Word level Language Identification," in *NLPAI*, December 2014. [Online]. Available: https://www.microsoft.com/en-us/research/publication/ye-word-kis-lang-ka-hai-bhai-testing-the-limits-of-word-level-language-identification/

[5] F. Xia, W. Lewis, and H. Poon, "Language ID in the context of harvesting language data off the web," in *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 2009, pp. 870–878.

[6] E. Tromp and M. Pechenizkiy, "Graph-based n-gram language identification on short texts," in *Proc. 20th Machine Learning conference of Belgium and The Netherlands*, 2011, pp. 27–34.

[7] A. Kunchukuttan, R. Puduppully, and P. Bhattacharyya, "BrahmiNet: A transliteration and script conversion system for languages of the Indian subcontinent," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 2015, pp. 81–85.

[8] M. Rosca and T. Breuel, "Sequence-to-sequence neural network models for transliteration," *arXiv preprint arXiv:1610.09565*, 2016.

[9] S. Sitaram and A. W. Black, "Speech synthesis of code-mixed text," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 3422–3428.

[10] S. Sitaram, S. K. Rallabandi, S. Rijhwani, and A. W. Black, "Experiments with Cross-lingual Systems for Synthesis of Code-Mixed Text." in *SSW*, 2016, pp. 76–81.

[11] K. R. Chandu, S. K. Rallabandi, S. Sitaram, and A. W. Black, "Speech Synthesis for Mixed-Language Navigation Instructions." in *INTERSPEECH*, 2017, pp. 57–61.

[12] A. L. Thomas, A. Prakash, A. Baby, and H. A. Murthy, "Code-switching in Indic speech synthesisers." in *Interspeech*, 2018, pp. 1948–1952.

[13] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[14] "Language independent phoneme mapping for foreign TTS, author=Badino, Leonardo and Barolo, Claudia and Quazza, Silvia, booktitle=Fifth ISCA Workshop on Speech Synthesis, year=2004."

[15] N. Campbell, "Talking foreign-concatenative speech synthesis and the language barrier," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[16] L. M. Tomokiyo, A. W. Black, and K. A. Lenzo, "Foreign accents in synthetic speech: development and evaluation," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[17] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Close to Human Quality TTS with Transformer," *CoRR*, vol. abs/1809.08895, 2018. [Online]. Available: http://arxiv.org/abs/1809.08895

[18] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning," *arXiv preprint arXiv:1907.04448*, 2019.

[19] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," *CoRR*, vol. abs/1712.05884, 2017. [Online]. Available: http://arxiv.org/abs/1712.05884

[20] W. Ping, K. Peng, A. Gibiansky, S. Ö. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: 2000-speaker neural text-to-speech," *CoRR*, vol. abs/1710.07654, 2017. [Online]. Available: http://arxiv.org/abs/1710.07654

[21] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. C. Courville, and Y. Bengio, "Char2Wav: End-to-End Speech Synthesis," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=B1VWyySKx

[22] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, 2019, pp. 3165–3174.

[23] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *CoRR*, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[24] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast High-Fidelity Speech Synthesis," *CoRR*, vol. abs/1711.10433, 2017. [Online]. Available: http://arxiv.org/abs/1711.10433

[25] J.-M. Valin and J. Skoglund, "LPCNet: Improving Neural Speech Synthesis Through Linear Prediction," 2018.

[26] H. Sun, X. Tan, J.-W. Gan, H. Liu, S. Zhao, T. Qin, and T.-Y. Liu, "Token-Level Ensemble Distillation for Grapheme-to-Phoneme Conversion," *arXiv preprint arXiv:1904.03446*, 2019.

[27] S. Gella, J. Sharma, and K. Bali, "Query word labeling and back transliteration for Indian languages: Shared task system description," *FIRE Working Notes*, vol. 3, 2013.

[28] R. S. Roy, M. Choudhury, P. Majumder, and K. Agarwal, "Overview of the FIRE 2013 track on transliterated search," in *Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation*, 2013, pp. 1–7.

[29] R. Grundkiewicz and K. Heafield, "Neural machine translation techniques for named entity transliteration," in *Proceedings of the Seventh Named Entities Workshop*, 2018, pp. 89–94.

[30] N. Chen, R. E. Banchs, M. Zhang, X. Duan, and H. Li, "Report of News 2018 named entity transliteration shared task," in *Proceedings of the seventh named entities workshop*, 2018, pp. 55–73.

[31] M. Choudhury, G. Chittaranjan, P. Gupta, and A. Das, "Overview and datasets of FIRE 2014 track on transliterated search," in *Preproceedings 6th workshop FIRE-2014*, 2014.