Learning pronunciation from a foreign language in speech synthesis networks

Younggun Lee[1]*, Suwon Shon*[2]*, Taesu Kim*[1]

[1]Neosapience, Inc., Seoul, South Korea
[2]Massachusetts Institute of Technology, Cambridge, MA, USA

{yg,taesu}@neosapience.com, swshon@mit.edu

Abstract

Although there are more than 65,000 languages in the world, the pronunciations of many phonemes sound similar across the languages. When people learn a foreign language, their pronunciation often reflects their native language's characteristics. This motivates us to investigate how the speech synthesis network learns the pronunciation from datasets from different languages. In this study, we are interested in analyzing and taking advantage of multilingual speech synthesis network. First, we train the speech synthesis network bilingually in English and Korean and analyze how the network learns the relations of phoneme pronunciation between the languages. Our experimental result shows that the learned phoneme embedding vectors are located closer if their pronunciations are similar across the languages. Consequently, the trained networks can synthesize the English speakers' Korean speech and vice versa. Using this result, we propose a training framework to utilize information from a different language. To be specific, we pre-train a speech synthesis network using datasets from both high-resource language and low-resource language, then we fine-tune the network using the low-resource language dataset. Finally, we conducted more simulations on 10 different languages to show it is generally extendable to other languages.

Index Terms: speech synthesis, cross-lingual, multilingual

1. Introduction

Among many languages in the world, some of the languages have phonemes with the same pronunciation. Conceptually, if the intersection of the pronunciations of two languages is large enough, the burden of learning the pronunciation of the one language after acquiring the other language will be lowered. We can also see people tends to show the accent of their first language when speaking in their second language [1]. We can also recognize easily that people tend to use their native language accent when speaking a foreign language. This may tell us that people learn new pronunciation based on their native language. Motivated by this, we investigated how the datasets from different languages can help training and generalization of speech synthesis model.

At first, we trained a bilingual TTS model using monolingual speakers' speech database. Deep neural network based Text-To-Speech (TTS) models allow us to build TTS models that can generate natural-sounding speech [2, 3, 4, 5]. Among the various approaches, end-to-end TTS models, such as Tacotron, require us only a little amount of prior knowledge about languages, and the TTS model can be trained easily if we have enough amount of text-speech pair data. Also, given enough amount of multiple speakers' speech data, we can build a multi-speaker TTS model when the amount of each speakers' speech data is relatively small [4]. In a similar way, we trained a multilingual multi-speaker TTS model using text-speech pairs of English and Korean. Interestingly, although we did not have any speaker who speaks both English and Korean, every speaker in the trained model could speak both English and Korean. To investigate how the model can generate a foreign language, we extracted phoneme embeddings and measured similarities between English and Korean phonemes. We found phonemes that have similar pronunciation tend to stay closer than the others even across the different languages. This result shows us that the interaction between two phonetic systems could help to generate foreign language using only the native language.

Based on this phonetic system interaction behavior, we hypothesized that the multilingual TTS model can be useful for training a TTS model of low-resource language. The major obstacles to train a TTS model for low-resource language is that lack of linguistic analysis components, such as dependency tree parser and morphological analyzer, and data itself [6]. Without linguistic analysis component, we cannot obtain linguistic features to train a TTS model. This problem can be circumvented with the help of end-to-end neural TTS models which require minimal prior knowledge. On the other hand, TTS models usually need a large amount of text-speech pair data for training. According to [7], original Tacotron requires more than 10 hours of speech data from one speaker to obtain good generation quality. It is reported that the model can benefit from pre-training Tacotron decoder with a large amount of speech data [7], but such large data is rarely available for the low-resource languages. Recall that, the learned phoneme embedding in multilingual TTS model tends to represent phoneme's pronunciation. Combining this observation with the pre-training approach led us to pre-train TTS model using a large amount of text-speech pair data from a high-resource language together with a small amount of text-speech pair from a low-resource language. The pre-trained model demonstrated higher performance in both subjective and objective measure. Moreover, we used the proposed approach to ten different languages to see if this method works for the other combination of languages.

To summarize, the contributions of this study are as follows:

1. We find two learned phoneme embeddings from different languages are located close if the pronunciation of the two phonemes is similar.

2. We show that pre-training a model with a large amount of data from high-resource language and target speaker's data can enhance the performance of the TTS model in low-resource language.

3. We validate the proposed pre-training framework by applying it to training TTS models of ten different languages.

2. Proposed method

2.1. Multilingual multi-speaker Tacotron

We implement a multilingual multi-speaker TTS model by modifying Tacotron [2]. We use simplified version [8] of Tacotron for the encoder and the decoder, but we use the original Tacotron style of Post-processing net and Griffin-Lim algorithm [9] for conversion of a linear-scale spectrogram to a waveform. A sequence of phonemes are converted to phoneme embeddings, then fed to the encoder as input. We concatenate phoneme embedding dictionary of each language to form the entire phoneme embedding dictionary, so there may exist duplicated phonemes in the dictionary if the languages share same phonemes. Note that, the phoneme embeddings are normalized to have the same norm. In order to model multiple speakers' voices in a single TTS model, we adopt Deep Voice 2 [4] style speaker embedding network. One-hot speaker identity vector is converted to a 32-dimensional speaker embedding vector by the speaker embedding network. Unless stated otherwise, we use the same hyperparameter settings with [8].

The model is trained to minimize L1 losses between ground-truth spectrogram and predicted spectrogram for both the linear-scale spectrogram and the Mel-scale spectrogram. When we train a multi-speaker TTS model, the amount of speech data differs across speakers. This data imbalance may induce a bias to the TTS model. To cope with this data imbalance, we divide the loss of each sample from one speaker by the total number of samples in a training set which belongs to the speaker. We empirically found that this adjustment in loss function yields better synthesis quality.

2.2. Multilingual pre-training

In the analysis of Subsection 4.2, having observed the learned phoneme embeddings from multilingual multi-speaker Tacotron, we noticed that two learned phoneme embeddings are located closer if they have similar pronunciation. We think that learning to generate speech in one language will help to learn to generate speech in another language because some information can be shared across the languages, such as phoneme-pronunciation relation of phonemes exist in common. We implemented this idea by pre-training a multilingual multi-speaker Tacotron with datasets from two languages.

We use the multilingual multi-speaker Tacotron for multilingual pre-training. Given a small amount of low-resource language text-speech pairs and a large amount of high-resource language text-speech pairs, we use the union of them to pre-train the model. There can be many ways to utilize information from other languages, such as decoder pre-training [7]. However, we chose to pre-train the entire model since it does not incur mismatch between pre-train and fine-tune as described in Section 3. Also, this model can potentially benefit from phoneme-pronunciation relation of languages learned in pre-training while decoder pre-training approach cannot use this information.

The pre-training is ceased when the validation loss stops to decrease. After that, the model is fine-tuned with the same text-speech pairs of the low-resource language.

3. Related work

For multilingual speech synthesis, some researchers used International Phonetic Alphabet (IPA), which can cover phones of all training languages, to transcribe texts of multilingual dataset [10, 11]. Alternative approaches other than using IPA were reported by taking a union of multiple languages' linguistic features [12] or defining new linguistic features [6]. However, these approaches still require prior knowledge of the low-resource languages to obtain linguistic features. End-to-end neural TTS models can lower the barrier of research and development of TTS models for low-resource languages since it requires only text-speech pairs. Also, additional information, such as grapheme-to-phoneme relation, can be easily integrated into the end-to-end neural TTS model to improve synthesis quality by replacing grapheme sequences with phoneme sequences.

Chung et al. proposed a data efficient semi-supervised training of TTS model that trains the model using a small amount of text-speech pair data with help of decoder pre-training [7]. They pre-trained decoder of Tacotron using only speech data while substituting the context vector with zero vector. From the pre-training, the decoder can learn how to generate acoustic representations, say Mel-spectrogram, in an auto-regressive way. After the pre-training, they fine-tuned the network with the target speaker's dataset. One shortcoming of this decoder pre-training is a mismatch in the distribution of the context vector between the training phase and the test phase. Also, the proposed pre-training approach inevitably assumes the existence of a large amount of speech data which may not be available for low-resource languages. Compared to this approach, our proposed method has lesser mismatch by using phoneme input during the pre-training, and it does not require large data for low-resource languages. Furthermore, our approach uses the target speaker's dataset at pre-train as well as at fine-tune. Since the target speaker's data is accessible at pre-training, there is no reason to avoid using it. It may increase the training time, but we think an improvement in generation quality by using the target speaker's data is more valuable than saving training time.

Other researchers were interested in transferring voice across languages [13], yet they only focused on transferring voice identity, not pronunciations of languages.

4. Experiments and results

4.1. Dataset

In Subsection 4.2, we used 60 hours of English data and 60 hours of Korean data. For English dataset, we used the union of VCTK dataset, CMU Arctic, and LJSpeech [14, 15, 16]. For Korean dataset, we used our proprietary multi-speaker dataset. In Subsection 4.3, we used the same Korean dataset and English dataset for pre-training and 2013 Blizzard Challenge dataset for an English target speaker's dataset. In Subsection 4.4, we additionally used CSS10 dataset which is the collection of single speaker speech datasets for ten different languages [17]. The ten languages include Chinese, Dutch, Finnish, French, German, Greek, Hungarian, Japanese, Russian, and Spanish.

We used grapheme-to-phoneme (G2P) libraries to convert text to the corresponding phoneme sequence. For English text, we used a G2P library [18] which extends CMUdict [19] phoneme dictionary by predicting phonemes of out-of-vocabulary words with a neural network. For Korean text, we used KoG2P [20] which is commonly used for Korean text preprocessing. We did not use G2P for CSS10 dataset, and we used pinyin transcript and romanized transcript for Chinese and Japanese respectively.

Table 1: *The 4-nearest English phonemes of each Korean phoneme whose pronunciation exists in English, IPA symbols are written in the parentheses*

Korean phoneme	1st	2nd	3rd	4th
h0 (h)	HH (h)	AH0 (ʌ)	AE0 (æ)	UW (u)
ii (i)	IY2 (i)	Y (j)	UW2 (u)	UW1 (u)
k0 (g)	G (g)	K (k)	DH (ð)	V (v)
kf (k)	K (k)	UW (u)	CH (tʃ)	G (g)
ll (l)	R (ɹ)	ER1 (ɝ)	ER2 (ɝ)	Y (j)
ng (ŋ)	M (m)	N (n)	NG (ŋ)	UH2 (ʊ)
p0 (b)	B (b)	OY0 (ɔɪ)	V (v)	P (p)
pf (p)	UW (u)	B (b)	P (p)	OY0 (ɔɪ)
qq (ɛ)	EY1 (eɪ)	EH1 (ɛ)	EY0 (eɪ)	ER1 (ɝ)
rr (ɾ)	IH0 (ɪ)	AE0 (æ)	UW (u)	AH0 (ʌ)
t0 (d)	D (d)	T (t)	TH (θ)	G (g)
tf (t)	UW (u)	T (t)	AE2 (æ)	D (d)
uu (u)	UW0 (u)	OY2 (ɔɪ)	OY1 (ɔɪ)	W (w)
vv (ʌ)	AO0 (ɔ)	AA0 (ɑ)	AH2 (ʌ)	AA2 (ɑ)

Table 2: *The 4-nearest Korean phonemes of each English phoneme whose pronunciation exists in Korean, IPA symbols are written in the parentheses, '(-)' denotes that there is no exact IPA symbol for that phoneme.*

English phoneme	1st	2nd	3rd	4th
AH0 (ʌ)	xx (ɯ)	rr (ɾ)	ps (-)	vv (ʌ)
AH1 (ʌ)	vv (ʌ)	ls (-)	wv (w)	wa (w)
AH2 (ʌ)	vv (ʌ)	lk (-)	lb (-)	aa (a)
B (b)	pp (p̤)	kk (k̤)	p0 (b)	tt (t̤)
D (d)	tt (t̤)	t0 (d)	cc (ts̤)	c0 (dz)
EH0 (ɛ)	ls (-)	lh (-)	ye (j)	aa (a)
EH1 (ɛ)	ya (j)	ee (e)	qq (ɛ)	aa (a)
EH2 (ɛ)	lm (-)	ya (j)	rr (ɾ)	qq (ɛ)
G (g)	kk (k̤)	k0 (g)	tt (t̤)	yq (j)
IY0 (i)	wi (w)	xi (ɯi)	ee (e)	ls (-)
IY1 (i)	wi (w)	yq (j)	xi (ɯi)	ii (i)
IY2 (i)	we (w)	ii (i)	wi (w)	ye (j)
K (k)	kh (k^h)	kk (k̤)	kf (k)	tt (t̤)
L (l)	vv (ʌ)	uu (u)	wv (w)	rr (ɾ)
M (m)	mf (m)	mm (m)	ng (ŋ)	nf (n)
N (n)	nn (n)	nf (n)	ng (ŋ)	mm (m)
P (p)	ph (p^h)	pp (p̤)	kh (k^h)	tt (t̤)
T (t)	th (t^h)	ch (ts^h)	tt (t̤)	ph (p^h)
UW (u)	pf (p)	tf (t)	rr (ɾ)	kf (k)
UW0 (u)	yu (-)	uu (u)	yo (-)	wv (w)
UW1 (u)	yu (-)	wi (w)	uu (u)	ii (i)
UW2 (u)	yo (-)	yu (-)	we (w)	wq (w)
W (w)	oo (o)	pf (p)	uu (u)	kf (k)
Y (j)	yq (j)	ii (i)	yu (-)	ll (l)

4.2. Training a bilingual multi-speaker TTS

We investigated how the phoneme representation is learned when the network is trained with the dataset containing both English and Korean.

It is known that the pronunciation of different languages can be described by one unified alphabetic system, IPA. The conversion tables of IPA-CMUdict and IPA-KoG2P can be found in [21] and appendix respectively. Although one-to-one correspondence does not hold between English phoneme set and Korean phoneme set in terms of IPA, we carefully chose a subset of each set to include only the phonemes that have the common pronunciations in both languages. The chosen subsets can be found in the appendix.

For each phoneme in the chosen subsets, anchor phoneme, we computed cosine distance between each of the other phonemes. Then, we listed the 4-nearest phoneme embedding of the opposite language to analyze the learned phoneme representation. The results are shown in Table 1 and Table 2. The numbers 0, 1, and 2 after the English phonemes denote "No stress", "Primary stress", and "Secondary stress" respectively. While CMUdict distinguished stressed pronunciation, we could not find corresponding IPA representation, so we reported all stress types. The results show that most of the anchor phonemes' corresponding phonemes in the opposite language were found in the 4-nearest phoneme embeddings. As shown in Table 2, the 4-nearest phonemes of 70 phonemes (57 phonemes in Korean) include the corresponding phonemes and similar pronunciations. It implies that the phoneme embeddings learned the relation of pronunciations across the languages.

Although not all of them have the corresponding pronunciation in the other language, pronunciation of the nearest phonemes sounded similar pronunciation. To check their perceptual similarity, we first generated speech of an English sentence, and we replaced each phoneme with the nearest Korean phoneme. Readers can find that both of the generated speeches sounds similarly in our demo page.[1]

4.3. Improved TTS model by multilingual pre-training

In this experiment, we hypothetically assumed English as a low-resource language since it is easier for readers to compare the quality of synthesized speech in English than other languages. We trained an English TTS model given a small amount of English dataset from a target speaker (Low-resource) and a large amount of Korean dataset (High-resource). Note that, we are not accessible to other English datasets since we are assuming English as a low-resource language. We first pre-trained TTS models to use information of other language and then fine-tuned them with low-resource target speaker data. There are three possible combinations of data for pre-training: a low-resource language only (equivalent with no pre-train), a high-resource language only, and union of them. We denoted the three ways of pre-train as T-base, PD-H, and PA-HL; details are explained in the following context.

The baseline, T-base, was not pre-trained at all. For PD-H (Pre-train Decoder with High-resource language), only the decoder module was pre-trained with 60 hours of Korean speech data following the method of [7]. PA-HL (Pre-train All modules

[1] http://neosapience.com/research/demo-learning-pron/

Table 3: *Result of preference test based on a 7-point rating scale.*

Data (hr)	Competing pair	Preference (%)		
		Former	Neutral	Latter
0.4	PD-E vs. PA-HL	29.7	16.3	**54.0**
2	PD-H vs. PA-HL	25.0	25.3	**49.7**
	PD-E vs. PA-HL	31.0	24.7	**44.3**
10	T-ori vs. PA-HL	20.3	11.3	**68.3**
	PD-H vs. PA-HL	26.0	29.3	**44.7**
	PD-E vs. PA-HL	33.0	22.7	**44.3**

Table 4: *Word error rate (%) by types of pre-training and amounts of data used for fine-tune.*

Model	Fine-tune data size (hr)		
	0.4	2	10
T-base	n/a	n/a	26.3
PD-H	n/a	37.4	21.1
PA-HL	55.8	**30.1**	**15.0**
PD-E	**41.9**	31.4	19.6

with High- and Low-resource language) was pre-trained with 60 hours of Korean text-speech pairs and the small English text-speech pairs of the target-speaker as described in Subsection 2.2. In addition to the three models, we compared a model PD-E which is pre-trained in the same way as PD-H except that we used 60 hours of English speech data instead of the Korean speech data. Note that, we may not be able to obtain PD-E in practice, since it can be difficult to find a large speech dataset for low-resource language. All of them have the same architecture described in Subsection 2.1.

After the pre-training, we fine-tuned each model with the English dataset of the target speaker. In the fine-tuning phase, we varied amounts of the target speaker's data used by 0.4, 2, and 10 hours. The fine-tuned models were compared through side-by-side preference test and calculating word error rate (WER). For the evaluation, we generated speech samples using 100 unseen test sentences. The test sentences were randomly selected from Harvard sentences [22]. We crowd-sourced workers using Amazon Mechanical Turk to compare speech samples from two different models. When calculating WER, we first fed generated speech samples to Google speech recognition API and calculated WER using the recognition result and the ground truth sentence. The result of the preference test and WER are summarized in Table 3 and Table 4 respectively, and the generated samples from each model are posted in the demo page.

During training, we observed that T-base and PD-H have difficulties to find attention alignment when the amount of data for fine-tuning is small. To be specific, T-base and PD-H failed to find attention for 0.4-hour fine-tune data, and only T-base failed when 2-hour fine-tune data were used. On the other hand, PA-HL and PD-E were successfully trained in all cases. From the result of Table 3, we can see that the raters preferred PA-HL in every competing pair. The proposed approach, PA-HL, could utilize information of the high-resource language for learning the low-resource language, and this led the model to show better generation quality. The WER of each approach in Table 4 shows a similar tendency with the preference test. We think the value of WER is higher than it sounds, probably because the speech recognition model had not been exposed to a synthesized speech during training. Still, we can compare the relative performance between models from the WERs.

Table 5: *Result of preference test based on a 7-point rating scale for models trained with CSS dataset.*

Language	Preference (%)		
	PD-H	Neutral	PA-HL
Chinese	25.0%	36.0%	**39.0%**
Dutch	28.0%	20.0%	**52.0%**
Finnish	30.0%	26.7%	**43.3%**
French	3.0%	6.0%	**91.0%**
German	14.0%	24.0%	**62.0%**
Greek	28.3%	13.3%	**58.3%**
Hungarian	25.6%	**37.8%**	36.7%
Japanese	25.0%	28.0%	**47.0%**
Russian	19.2%	20.8%	**60.0%**
Spanish	20.0%	20.0%	**60.0%**

4.4. Multilingual pre-training for other languages

We validated our approach by applying the same pre-training framework in other languages. We repeated the same experiment which is done in Subsection 4.3 with datasets different language. For each language, we used 2 hours of data to simulate low-resource language and compared PA-HL to PD-H by preference tests. Table 5 shows the result of the preference test of each language. While the performance gaps varied from language to language, PA-HL outperformed in most of the languages as it was in the previous subsection. We conjecture that the varying performance gap came from the difference in the phonology of each language, and we will investigate it in the future works.

5. Conclusion

In this work, we have trained multilingual multi-speaker TTS models using pairs of two monolingual datasets. By investigating the distance between learned phoneme embeddings, we have experimentally shown that the embeddings can represent the relation of pronunciations across the different languages. We proposed a pre-training framework that utilizes information of high-resource language to help training of low-resource language TTS model. The proposed approach outperformed other pre-training frameworks as well as the baseline Tacotron. Furthermore, by applying this pre-training framework to other 10 languages, we validated that the proposed method is generalizable to other pairs of languages.

Also, we are interested in applying the findings of this work to train an automatic speech recognition (ASR) system of low-resource languages. Like TTS, ASR also requires a large amount of training data. We may train a TTS model as proposed in this work. Since the TTS model can generate speech of arbitrary sentence in various speakers' voice, it will improve the robustness of the ASR model. We hope to investigate how the multilingual TTS can help training of the ASR model in the future.

6. References

[1] J. E. Flege, C. Schirru, and I. R. MacKay, "Interaction between the native and second language phonetic subsystems," *Speech communication*, vol. 40, no. 4, pp. 467–491, 2003.

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

[3] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman *et al.*, "Deep voice: Real-time neural text-to-speech," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 195–204.

[4] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 2962–2970.

[5] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Neural speech synthesis with transformer network," *arXiv e-prints*, 2018.

[6] A. Gutkin, "Uniform multilingual multi-speaker acoustic model for statistical parametric speech synthesis of low-resourced languages," in *Proc. of Interspeech 2017*, August 20–24, Stockholm, Sweden, 2017, pp. 2183–2187.

[7] 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," *ArXiv e-prints*, 2018.

[8] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, 2018, pp. 4779–4783.

[9] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time fourier transform," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '83, Boston, Massachusetts, USA, April 14-16, 1983*, 1983, pp. 804–807.

[10] 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 Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1713–1724, Aug 2012.

[11] I. P. Association, C. PRESS, and D. Decker, *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*, ser. A Regents publication. Cambridge University Press, 1999.

[12] B. Li and H. Zen, "Multi-language multi-speaker acoustic modeling for lstm-rnn based statistical parametric speech synthesis," 2016.

[13] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *arXiv preprint arXiv:1806.04558*, 2018.

[14] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2016.

[15] J. Kominek, A. W. Black, and V. Ver, "Cmu arctic databases for speech synthesis," Tech. Rep., 2003.

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

[17] K. Park and T. Mulc, "Css10: A collection of single speaker speech datasets for 10 languages," https://github.com/Kyubyong/css10/, 2018.

[18] K. Park and J. Kim, "g2p-en," https://github.com/Kyubyong/g2p, 2018.

[19] A. I. Rudnicky, "The cmu pronouncing dictionary," http://www.speech.cs.cmu.edu/cgi-bin/cmudict, 2015.

[20] Y. Cho, "Kog2p," https://github.com/scarletcho/KoG2P, 2017.

[21] L. Rice, "Hardware and software for speech synthesis," *Dr. Dobbs J. of Computer Calisthenics & Orthodontia*, vol. 1, no. 4, 1976.

[22] E. H Rothauser, W. D Chapman, N. Guttman, H. R Silbiger, M. Hecker, G. E Urbanek, K. S Nordby, and M. Weinstock, "Ieee recommended pratice for speech quality measurements," *scshape Ieee Trans. on Audio and Electroacoustics*, vol. 17, pp. 225–246, 09 1969.