INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Analysis of Pronunciation Learning in End-to-End Speech Synthesis

Jason Taylor, Korin Richmond

[1]Centre for Speech Technology Research, The University of Edinburgh
{jason.taylor, korin.richmond}@ed.ac.uk

Abstract

Ensuring correct pronunciation for the widest possible variety of text input is vital for deployed text-to-speech (TTS) systems. For languages such as English that do not have trivial spelling, systems have always relied heavily upon a lexicon, both for pronunciation lookup and for training letter-to-sound (LTS) models as a fall-back to handle out-of-vocabulary words (OOVs). In contrast, recently proposed models that are trained "end-to-end" (E2E) aim to avoid linguistic text analysis and any explicit phone representation, instead learning pronunciation implicitly as part of a direct mapping from input characters to speech audio. This might be termed *implicit LTS*. In this paper, we explore the nature of this approach by training *explicit LTS* models with datasets commonly used to build E2E systems. We compare their performance with LTS models trained on a high quality English lexicon. We find that LTS errors for words with ambiguous or unpredictable pronunciations are mirrored as mispronunciations by an E2E model. Overall, our analysis suggests that limited and unbalanced lexical coverage in E2E training data may pose significant confounding factors that complicate learning accurate pronunciations in a purely E2E system.

Index Terms: Speech Synthesis, End-to-End, Letter-to-Sound, Grapheme-to-Phoneme

1. Introduction

General-purpose TTS systems must ensure correct pronunciation for wide ranging text input. This is difficult as English orthography is often ambiguous, with complex letter-to-sound relationships and common non-standard words such as foreign names and loan words, homographs, number ambiguities and abbreviations [1].

The typical solution, used in systems like Festival [2], Mary [3] or Sparrowhawk [4], is to store the pronunciation of a finite list of known words in a lexicon which may be used for lookup. The lexicon may then be further used to disambiguate these pronunciations with modules like POS taggers and homograph disambiguators [5]. An additional function of the lexicon is to provide training data for a statistical letter-to-sound (LTS) predictor, also known as a grapheme-to-phoneme (G2P) model, for out-of-vocabulary words (OOVs). LTS has a long history in English TTS [6], and neural sequence-to-sequence models are the current state of the art [7].

Recently however, monolithic neural network based TTS systems have been proprosed that are trained end-to-end (E2E), like Tacotron 2 [8] or Deep Voice 3 [9], which jointly learn the traditional front-end steps (linguistic analysis) and back-end processing (waveform generation) simultaneously. These function without a lexicon or LTS model. The simplicity of building an E2E model is attractive for developing TTS systems in new languages for which costly front-end tools do not exist.

E2E models must learn to generalise from character input to acoustic output from fairly large sets of parallel text and speech audio data, and in this way implicitly learn pronunciation knowledge. Publicly available single-speaker sets, such as the Linda Johnson (LJ) [10] and the Nancy Blizzard [11] corpora, or industry-only datasets in Tacotron 2 and DeepVoice 3, contain 15-25 hours of speech. Meanwhile, the open-source multi-speaker VCTK corpus [12] contains 44 hours of speech.

Table 1: *Lexical coverage in large TTS datasets*

Datum	LJ	Nancy	VCTK
Total Word Types	14,750	18,695	5,839
Total Word Tokens	225,715	170,018	326,971
Total Sentences	13,100	12,095	44,070
Total Length (hours)	24	17	44
Mean Utt Length (words)	17.2	14.1	7.4

Yet despite their long lengths in terms of total hours of speech, these data sets in fact contain only relatively narrow coverage of unique words. As shown in Table 1, each corpus contains a large number of words overall (tokens) but a low number of unique words (types) compared with the size of common pronunciation lexica: approximately 135,000 in the CMUdict [13], 165,000 in Unisyn [14] and 145,000 in Combilex [15]. This suggests pronunciation modelling by an E2E system could well be weaker than in an LTS model trained with the far broader coverage found in a lexicon. Indeed, Tacotron 2 researchers noted mispronunciations were a common cause of errors: out of 100 utterances, 6 contained incorrect phonentic pronunciations, and a further 23 contained prosody errors including incorrect lexical stress placement [8]. One can enhance E2E model input with linguistic information created by a separate front-end, such as phones, stress and syllabification [16], but the resulting system would arguably no longer be truly E2E since it would function primarily as a back-end only.

We propose in this paper to explore the nature and extent of pronunciation learning in purely E2E-trained systems. We do this in part by training state-of-the-art neural network-based LTS models on lexical data alone, as a sort of proxy model. Specifically, we choose different lexical sets for comparison. These range for example from a full lexicon typically used to train an LTS model to a more limited set comprising an E2E training corpus transcript, then to a very much more restricted set of only the unique words contained in that corpus transcript. These experiments demonstrate what an explicit LTS model is able to learn from the different sets of words available under the different conditions. However, to relate these results to pronunciation learning in E2E TTS systems, we also need to establish whether explicit LTS models trained with the words contained in E2E training sets are indeed informative of an E2E system's implicit pronunciation model. To achieve this, we compare errors made by the explicit LTS models to corresponding audio samples synthesised by an E2E model trained on the data.

http://dx.doi.org/10.21437/Interspeech.2019-2830

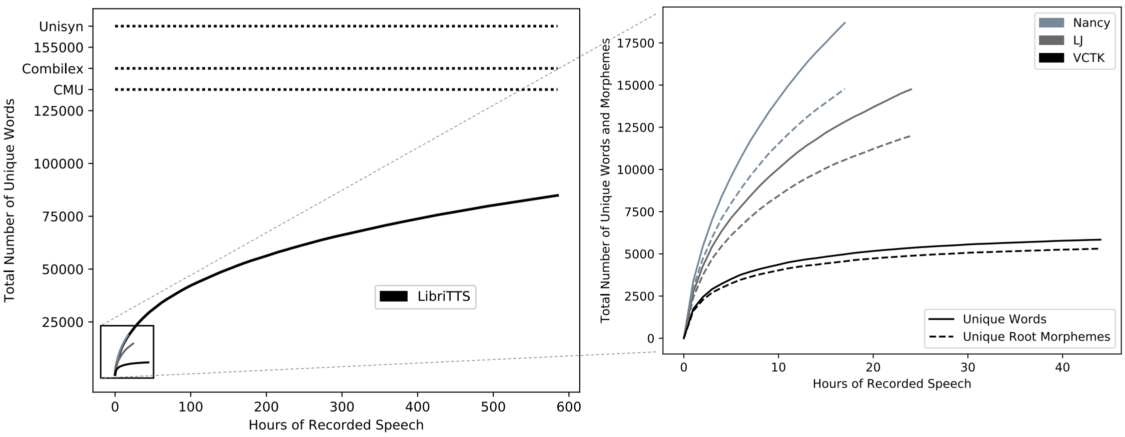

Figure 1: *Cumulative coverage of words per dataset*

Though the concept at the heart of this paper is relatively straightforward, practical constraints make completely equal objective comparison somewhat difficult. One significant complication, for example, concerns the different domains in which pronunciations must be evaluated. Explicit LTS models predict a sequence of discrete phone symbols for a given word, whereas E2E TTS models aim to predict an accurate speech waveform. During training the latter learns non-symbolic pronunciation representations that yield continuous speech sounds with appropriate prosody at synthesis time. The perception of pronunciation from an E2E model is therefore inextricably linked to this acoustic model. Whereas explicit LTS models can be evaluated with Word Error Rate (WER) and Phone Error Rate (PER), objective measures of TTS quality do not directly measure pronunciation accuracy.

Despite such methodological difficulties inherent in comparing LTS and E2E model output, however, this paper highlights coherent patterns amongst the types of words which cause errors in both systems. These observations serve as valuable insight into the potential for E2E-trained systems to learn an adequate pronunciation model, and how performance might depend on the lexical coverage of the training data.

2. Lexical Coverage in TTS Datasets

We perform LTS experiments on the LJ, Nancy, and VCTK corpora. The recording script for these 3 data sets determines the coverage of words that an E2E model will use to learn pronunciation. The LJ corpus contains utterances from 7 non-fiction books in the public domain about history, arts and cookery. The Nancy corpus contains a wide variety of text algorithmically selected to maximise phonetic coverage. The VCTK corpus contains 400 utterances selected from the Glasgow Herald and the Rainbow Passage recorded by 108 speakers. The lexical coverage of the 3 data sets, shown cumulatively in random-sentence order, is shown in Figure 1. The steepness of each curve demonstrates the increasing lexical coverage per hour of each dataset. Notably, they exhibit the text selection method, for instance VCTK has a relatively flat curve since many utterances were repeated whilst Nancy's curve is steep due to an attempt to maximise the number of phonetic contexts contained in the script. Lexical diversity may also depend on other factors such as the genre of the text and the total number of hours.

Theoretically, one could accumulate more and more speech to achieve the same phonetic coverage as a lexicon. However, the natural frequency of words tends to follow a Zipfian distribution, meaning the number of new words added per hour of speech data progressively flattens. This is shown by the curve for the recently announced LibriTTS dataset [17] on the left side of Figure 1, which contains 585 hours but still has lower phonetic coverage than a lexicon, despite the substantial financial and computational costs of collecting that much data for training a TTS system. A lexicon thus provides more comprehensive lexical coverage, and obtaining comparable coverage from speech corpora requires an exponential increase in audio data, which becomes increasingly costly and problematic.

3. Experiments

3.1. Explicit LTS Data

We create training data sets for explicit LTS models from the E2E training corpora by phonetizing the text using the General American surface-form of Combilex. Figure 2 presents a pseudo-Venn diagram of lexical content compared with Combilex. LJ and Nancy contained a substantial number of OOVs, indicated by the numbers outside of the Combilex circle. In principle, these could be transcribed, either manually or using a Combilex LTS model. However, we are here simply exploring the viability of this approach and looking at general error patterns, rather than computing precise error rates. We have therefore not introduced transcriptions for these OOVs, and have trained the explicit LTS models without them. This means in effect that approximately 10% more data was used to train the E2E models when using LJ or Nancy than was available to the explicit LTS models. Note that though this may tend to disadvantage the explicit LTS models, it does not have a significant effect on our analysis below.

As a first processing step, all characters were lower-cased throughout, as in the training of the E2E model. Single pronunciations were selected for homographs and digits were excluded since disambiguation of these words is a separate front-end task in itself [5]. Foreign names and loan words were included as these tend to exhibit unusual LTS relationships. All punctuation except hyphens denoting compound words and apostrophes denoting possessive *s* was removed.

The training data patterns are presented to the neural networks in our experiments variously as either i) isolated unique words, ii) isolated word tokens, or iii) whole-utterance sequences. The validation and test words were in contrast always predicted in isolation and consisted of words present in Combilex but not found in any of the training sets. Naturally, the

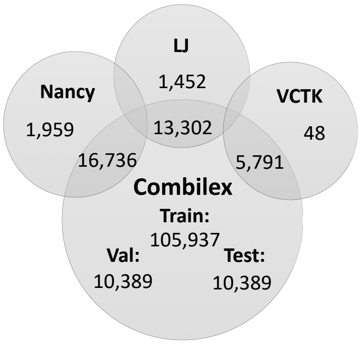

Figure 2: *Words shared and not shared between Combilex and the E2E TTS training sets. NB: overlap also exists between LJ, Nancy and VCTK, but this is not shown to ensure clarity.*

most frequent words in English, such as closed class and other common words, are contained in the E2E training sets. This leaves less common words for the validation and test sets. Note, we have also deemed it necessary to test the model trained on entire sentences on single words, even though it may not be optimised for these sequence lengths. This is to enable comparison between the performance of the LTS models trained on sequences of single and multiple words.

3.2. Explicit LTS Models

We use the OpenNMT [18] package in PyTorch, originally developed for Machine Translation, to train Bidirectional Long Short Term Memory (BLSTM) models for LTS model building. The relevant hyper-parameters are: 6 bi-directional encoder and decoder layers with 500 units each, a learning rate of 0.0001, dropout of 0.1, Luong's global attention [19], the ADAM optimiser and mini-batches of 64. The BLSTMs tend to converge after between 50,000 and 100,000 training steps.

3.3. E2E Model

To demonstrate the implicit LTS model in an E2E TTS system is learning similar information about pronunciation as our explicit LTS models, we synthesise a selection of the explicit LTS model test set with an E2E model trained on the full data sets. All utterance waveforms are first downsampled to 16kHz for this purpose. We then use an open-source implementation of Deep-Convolutional_TTS [20]. This is composed of two neural sequence models: Text-to-mel (with global attention), and the Super Spectrogram Resolution Network (SSRN) which refines coarse mel spectrograms to full spectrograms. These were trained for 300 epochs each and together form the implicit LTS model. Griffin-Lim was used to re-introduce phase, and so generate synthetic speech samples.

4. Results

4.1. Explicit LTS

We can compare LTS results of models trained on Combilex and the E2E recordings to reason about the quality of the pronunciation modelling in E2E systems. Table 2 shows the results of the explicit LTS models with the E2E datasets. The total number of incorrect phone strings divided by the size of the test set gives the Word Error Rate (WER), which indicates how many words in total are incorrectly predicted. The Phone Error Rate (PER) is calculated by summing the total Levenstein distance for every predicted sequence in the test set and dividing by the sum of the lengths of all reference phone strings. This quantifies how many phones out of 100 are wrong.

Table 2: *LTS of E2E training recordings*

Input Type	Metric	LJ	Nancy	VCTK
Types	**WER**	52.9%	44.4%	82.5%
	PER	13.6%	10.3%	30.3%
Tokens	**WER**	64.0%	60.1%	89.5%
	PER	19.7%	17.7%	38.7%
Sequences	**WER**	42.5%	37.9%	57.7%
	PER	10.7%	8.9%	14.9%

When Combilex's training data (105,937 entries) is used, the LTS models perform with PER=1.1% and WER=4.9%. The error rates for the E2E datasets shown in the top two rows of Table 2 are higher. This is because the neural sequence-to-sequence models are trained with fewer unique word types (and morphemes) than are contained in Combilex. It is particularly high for VCTK which has less than 6,000 word types. Models trained on the Nancy corpus, with the highest number of unique word types, performs best of the three E2E training data sets.

The middle two rows of Table 2 show that error rates when training on all word tokens are higher than the unique word type error rates. This is presumably because the LTS model becomes biased towards frequent word types when trained on data with very unbalanced coverage (following the well-known Zipf distribution characteristic of human language). This negative effect is very much likely to be reflected in the E2E model, as its training data likewise reflects the natural frequencies of words.

The error rates for models trained on word sequences (bottom two rows of Table 2) are the lowest. This suggests that longer sequences lead to better results in the LTS model. We trained further LTS models on the Nancy data with an increasing number of tokens per sequence to test this assumption. The tokens were initially combined in the order they appear in the recordings, but randomising the tokens in the sequences makes little difference, suggesting implicit word-level language model information does not help LTS learning. These results are presented in Figure 3, with highly variable WER for sequences of 2, 3, 4 and 5 tokens before levelling out for sequences of more tokens. The result suggests input and output sequences of at least 60 characters, or 6 words, are optimal to train the explicit LTS model on the training recordings. The E2E models are all trained on sequences with an average of more than 7 words (see Table 1), although the variable error rates with lower tokens per sequence may be particular to the model architecture we use here. The effect of sequence length is difficult to tease apart from other characteristics of the data set, since permuting the order of utterance audio is non-trivial.

One problem in LTS evaluation is that objective error rates do not reflect the plausibility of predicted sequences which are similar to but different from the true reference sequences. For instance, even if the letter *i* in the word *tamil* were predicted as [*I*] (in Combilex's symbolic representation), which would not be a terrible error, it is entered as the schwa phone [@] in the test set and so would count as a word and phone error. However, modifying the metrics to account for plausible alternatives, as attempted in [21], is not straightforward, as the correct pronunciation is not easily predictable from the letters: the letter *y* in *flytraps* was predicted as the monothong [*I*] whereas the correct pronunciation is the diphthong [*aI*]. Other such inappropriate

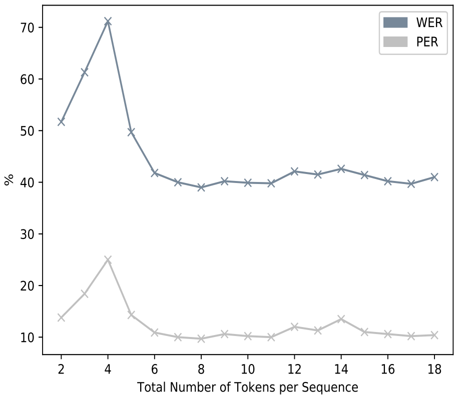

Figure 3: *LTS results for different training sequence lengths.*

alternatives include the pronunciation of *ph* across a morpheme boundary as the voiceless labiodental fricative symbol, [*f*], in *loophole*. To interpret the explicit models as a reflection of implicit LTS models, we should recognise the E2E model does not deal with this intermediate phonetic representation directly, and hence is not affected by this. Practical LTS performance is likely to be higher than the error metrics suggest.

4.2. E2E Synthesis of Explicit LTS Error Words

As motivated previously in Section 1, it is important to establish the validity of relating explicit LTS performance to pronunciation modelling in an E2E system. We therefore synthesised words which were predicted correctly and incorrectly from the LJ sequence LTS model. One hundred correctly predicted words were randomly extracted, and a further 100 incorrectly predicted words were hand-selected according to 4 categories: i) cases where LTS gives plausible alternatives (e.g. *tamil* above); ii) where LTS predicts inappropriate alternatives (e.g. *loophole*); iii) English names with difficult orthography (e.g. *Loughborough*); and iv) foreign names or loan words (e.g. *Flaubert* and *karate*). Synthesising isolated words in their citation form is sub-optimal for the E2E model, as it is trained on longer sequences. Therefore, we embedded the test words into a carrier sentence: '*Now we will say __ again.*'

We do not seek here to judge the overall quality of an E2E system against an equivalent that uses a lexicon. Rather, we aim to demonstrate that similar issues arise in implicit and explicit LTS pronunciation models. Moreover, we highlight that explicit knowledge of a pronunciation is often necessary to ensure correct synthesis. Of the 100 words predicted correctly by the LTS model, 79 were understood by our listener and 21 were unrecognised. Many of these words were difficult to understand without context (e.g. *flutings* and *sluicing*), with slight mispronunciations rendering the words unrecognisable. The listener often mistook them for more common words, for instance *mesher* was misunderstood as *measure*. Despite these perceptual difficulties, the overall trend is that the words predicted correctly by the explicit LTS model are also intelligible to our listener when synthesised by the E2E system.

For the G2P error words, we also asked whether pronunciations were correct or not. This was to reflect that not all G2P errors are equal. Words with plausible but formally incorrect G2P predictions were synthesised more intelligibly on the whole. This is shown by the *Plausible LTS* column in Figure 4, where the "Unrecognisable" bar is shorter than for the other categories. Specifically in this category, unrecognisable pronunciations produced by the E2E model were caused by stress placement on incorrect syllables. For example, the first syllable was more prosodically salient than the second in *regina*.

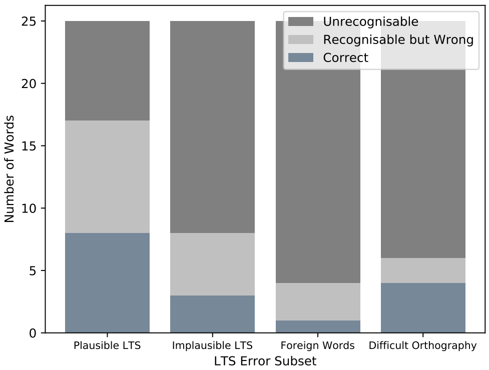

Figure 4: *Expert listener judgement on E2E TTS for LTS error words.*

Looking at the other 3 categories of LTS model error words, it is striking that they are associated with larger proportions of unrecognisable E2E pronunciations. Samples of such words with their pronunciations and audio are available online[1]. Examples include [in IPA]: *Siobhan* [siəʊbæn]; *Loughborough* [ləʊbərəʊ]; *anchoring* [əʧɔrɪŋ].

Overall, there are clear correspondences in pronunciation modelling between an explicit LTS model and our E2E TTS system. In short, words that are correctly predicted by the LTS model also tend to be correctly pronounced by the E2E TTS voice. Conversely, words whose pronunciations are incorrectly predicted by the LTS model also tend to be incorrectly pronounced by the E2E model. Moreover, this relationship is graded: words with less serious phone prediction errors tend to be pronounced more successfully by the E2E voice too, whereas words that are more obviously errorprone for the LTS models are likewise more likely to be mispronounced by the E2E voice. This suggests that while it is difficult to quantify the exact accuracy of pronunciation learning, the nature of the data used to train E2E models will determine the effectiveness of the implicit pronunciation model that is learned.

5. Conclusions

This paper presents an analysis of pronunciation modelling in end-to-end TTS. We compare explicit LTS models trained using a lexicon to equivalent models trained using text from typical data sets used to train end-to-end (E2E) TTS systems. We find the diversity and balance of lexical coverage of these data sets to be significantly lower than that of a lexicon. Our results suggests this negatively impacts the ability of an E2E model to learn correct pronunciations. These factors should therefore be taken into account when selecting data to train end-to-end TTS systems, or alternatively, other strategies for ensuring correct pronunciations may need to be considered.

[1] Audio samples at: http://homepages.inf.ed.ac.uk/s1649890/lts/

6. References

[1] P. Taylor, *Text-to-Speech synthesis*. Cambridge: Cambridge University Press, 2009.

[2] R. A. J. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[3] M. Schröder and J. Trouvain, "The German text-to-speech synthesis system MARY: a tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003.

[4] P. Ebden and R. Sproat, "The Kestrel TTS text normalization system," *Natural Language Engineering*, vol. 21, no. 3, p. 333353, 2015.

[5] K. Gorman, G. Mazovetskiy, and V. Nikolaev, "Improving homograph disambiguation with supervised machine learning," in *LREC 2018 - Language Resources and Evaluation Conference, Proceedings*, 2018.

[6] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2006.

[7] K. Yao and G. Zweig, "Sequence-to-Sequence Neural Net Models for Grapheme-to-Phoneme Conversion," *CoRR*, vol. abs/1506.00196, 2015. [Online]. Available: https://arxiv.org/pdf/1506.00196.pdf

[8] J. Shen *et al.*, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Proceedings*, 2018.

[9] W. Ping *et al.*, "Deep Voice 3: scaling text-to-speech with convolutional sequence learning," in *ICLR 2018 - International Conference on Learning Representations, Proceedings*, 2018.

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

[11] CSTR, "The Nancy corpus," 2011. [Online]. Available: http://www.cstr.ed.ac.uk/projects/blizzard/2011/lessac_blizzard2011/

[12] C. Veaux, J. Yamagishi, and K. MacDonald, "VCTK corpus: English multi-speaker corpus," 2019. [Online]. Available: https://datashare.is.ed.ac.uk/handle/10283/2651

[13] CMU, "The Carnegie Mellon pronouncing dictionary," 2018. [Online]. Available: https://github.com/cmusphinx/cmudict

[14] S. Fitt, "Unisyn lexicon," 2018. [Online]. Available: http://www.cstr.ed.ac.uk/projects/unisyn/

[15] K. Richmond, "Combilex speech technology lexicon," 2018. [Online]. Available: http://homepages.inf.ed.ac.uk/korin/sitenew/Research/Combilex

[16] K. Kastner *et al.*, "Representation mixing for TTS synthesis," in *ICASSP - IEEE International Conference on Acoustics, Speech and Signal Processing, Proceedings*.

[17] H. Zen *et al.*, "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech," in *Submission to Interspeech*, 2019. [Online]. Available: https://arxiv.org/abs/1904.02882

[18] G. Klein *et al.*, "OpenNMT: Open-source toolkit for neural machine translation," in *ACL 2017 - Annual Meeting of the Association for Computational Linguistics, Proceedings*, 2017.

[19] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *ACL 2015 - Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Proceedings, 2015.

[20] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Proceedings*, 2018.

[21] B. Hixon, E. Schneider, and S. L. Epstein, "Phonemic similarity metrics to compare pronunciation methods," in *Interspeech 2011 - Annual Conference of the International Speech Communication Association, Proceedings*, 2011.