Investigating the robustness of sequence-to-sequence text-to-speech models to imperfectly-transcribed training data

Jason Fong, Pilar Oplustil Gallegos, Zack Hodari, Simon King

The Centre for Speech Technology Research, University of Edinburgh, United Kingdom
{jason.fong, p.s.oplustil-gallegos, zack.hodari, Simon.King}@ed.ac.uk

Abstract

Sequence-to-sequence (S2S) text-to-speech (TTS) models can synthesise high quality speech when large amounts of annotated training data are available. Transcription errors exist in all data and are especially prevalent in found data such as audiobooks. In previous generations of TTS technology, alignment using Hidden Markov Models (HMMs) was widely used to identify and eliminate bad data. In S2S models, the use of attention replaces HMM-based alignment, and there is no explicit mechanism for removing bad data. It is not yet understood how such models deal with transcription errors in the training data.

We evaluate the quality of speech from S2S-TTS models when trained on data with imperfect transcripts, simulated using corruption, or provided by an Automatic Speech Recogniser (ASR). We find that attention can skip over extraneous words in the input sequence, providing robustness to insertion errors. But substitutions and deletions pose a problem because there is no ground truth input available to align to the ground truth acoustics during teacher-forced training. We conclude that S2S-TTS systems are only partially robust to training on imperfectly-transcribed data and further work is needed.

Index Terms: speech synthesis, sequence-to-sequence models, found data

1. Introduction

Sequence-to-sequence (S2S) text-to-speech (TTS) models [1, 2, 3] using an encoder-decoder-with-attention architecture [4] can produce speech of near-human quality, given large amounts of accurately-transcribed speech data. Training with over 24 hours of data can produce voices with almost perfect naturalness [2]. While it is possible to build an *intelligible* voice with just 3 hours of data [5], such little data not only limits overall naturalness but also increases the rate of gross synthesis errors [6].

One option to increase the amount of training data is to pool multiple speakers, using speaker embeddings to account for the additional variation [3, 7, 6]. Another is to use unpaired text or audio to pre-train the model [5].

Here we investigate the more obvious option of found data: transcribed speech that was not originally intended for TTS, available in vast quantities from audiobooks, podcasts, etc. But found data contains more errors than TTS-specific data. Text-speech mismatches arise from inaccurate human transcribers or Automatic Speech Recognition (ASR), and speaker error.

For older TTS paradigms there are methods for dealing with transcription error (Section 2) but it is not clear how robust S2S-TTS models are to imperfect data. The attention mechanism is responsible for alignment during training, and for duration prediction during synthesis. This mechanism has the *potential* to align text and audio in a non-monotonic fashion and thus skip over errors (unlike standard forced alignment which can only align incrementally). However, whatever the attention mechanism learns to do for training data may *also* occur during synthesis, where skipping words is probably undesirable. To gain some insight, we investigate S2S-TTS models trained on transcripts corrupted in a variety of ways.

2. Related work

Previous work has investigated the impact of building unit-selection and HMM-based statistical parametric speech synthesis (SPSS) systems on imperfect data. Whilst data cleaning techniques are applicable to all paradigms [8, 9], each also has its own mechanisms to mitigate the effects of transcription errors.

In **unit selection**, the forced alignment used to annotate the data can identify transcription mismatches [10]. At run-time, the join cost or acoustically-based target cost (in **hybrid unit selection**) reduce the chance of selecting units with acoustic properties that do not correspond to their label. The quality of unit selection is dramatically reduced by such annotation errors because resulting phone-level mis-retrievals cause local unintelligibility [11].

In **HMM synthesis**, beam pruning during Baum-Welch estimation can prevent the model learning from mistranscribed parts of the data. HMM-based SPSS systems have also been shown to be more robust than unit selection systems when using phonetically-imbalanced corpora recorded in suboptimal conditions [12].

The attention mechanism is an essential component of S2S models. For each decoder step, attention learns to align ('attend') to the relevant information in the input. First applied in machine translation, attention allows for non-monotonic alignment between input and output sequences, which is essential in languages that have different word orders [4].

In contrast to previous TTS paradigms or even the more recent WaveNet [13] model, S2S models with attention can learn a non-monotonic alignment between text and audio. Specifically, irrelevant items in the input sequence will never be attended to: they will be 'skipped over'.

3. Data

In all experiments we use the LJ Speech data set[1] which comprises ~24 hrs of audio in 13,100 sentences from a single American female speaker reading 7 non-fiction LibriVox books. Normalisation of numbers, ordinals, and monetary units is already applied to the transcriptions, but non-standard words such as acronyms are not normalised. The text contains punctuation such as commas and periods. We perform only superficial text preprocessing: removing capitalisation and inserting tokens for

[1] https://keithito.com/LJ-Speech-Dataset/

word boundaries, punctuation, and the start/end of sentences. Average sentence length is 17 words (100 characters).

4. Simulating Transcription Errors

To control the experimental conditions of our investigation, we deliberately corrupt LJ Speech transcriptions to simulate manual and automatic annotation errors.

4.1. Simulating errors in manual transcription

We devised three methods for corrupting the text of a sentence that simulate word-level transcriber (or speaker) errors.

- **Add 5 words**: 5 words (length [-1,+1] around the mean 7) from the vocabulary are inserted at random positions.
- **Delete 5 words**: 5 random words are deleted from each sentence (but leaving a minimum of 1 word).
- **Replace 5 words**: 5 random words are each replaced by a word of equal length sampled from another sentence.

4.2. Simulating errors in automatic transcription

We ran the audio for each utterance through a typical HMM–DNN ASR system, and took the top 50 hypotheses from the lattice. We calculated the word error rate (WER) for each of the 50 hypotheses and chose that with the highest WER per utterance. ASR errors include insertions, deletions, and substitutions. In contrast to the method in Section 4.1, ASR errors are acoustically-plausible; e.g., "in being comparatively modern" → "indie comparatively modeled".

5. System description

Our model is a modified version of the deep-convolutional TTS system (DC–TTS) [14], a convolutional encoder-decoder with an autoregressive structure in the decoder[2].

To simplify training, we follow standard DC–TTS procedure by training two separate models independently on the ground truth. The S2S portion of the model, called text-to-Mel (T2M), predicts 'coarse' 80-band Mel filter banks (MFB) from graphemes. This intermediate representation is extracted from the ground truth magnitude spectrogram (80 frames per second) using; pre-emphasis, Mel-scale bins, and timescale reduction by factor of 4 (resulting in a frame rate of 20). Following T2M, the spectrogram super-resolution network (SSRN) uses transposed convolutions to reconstruct the full-resolution magnitude spectrogram. Placed in sequence, these two models perform grapheme to magnitude spectrogram prediction.

DC–TTS contains 4 trainable modules, shown in blue in Figure 1. The T2M model comprises three modules that together predict the coarse MFBs $\hat{Y}$, and are trained separately from SSRN.

- TEXTENC: Encodes N one-hot characters to a sequence of d dimensional keys K and values V for use in attention querying (normally, attention learns a projection $V \rightarrow K$).
- AUDIOENC: Encodes t coarse MFB frames to d dimensional queries $Q_{0:t}$, where t is the current decoder timestep.
- ATTENTION: Multiplicative attention [15] which determines the alignment $A_{0:t} = softmax(K^\top Q_{0:t} * \sqrt{d})$ used to calculate the result $R_{0:t} = A_{0:t}V$. For synthesis, attention is forced to be monotonic and can only skip forward a maximum of 3 encoded characters per decoder time-step.

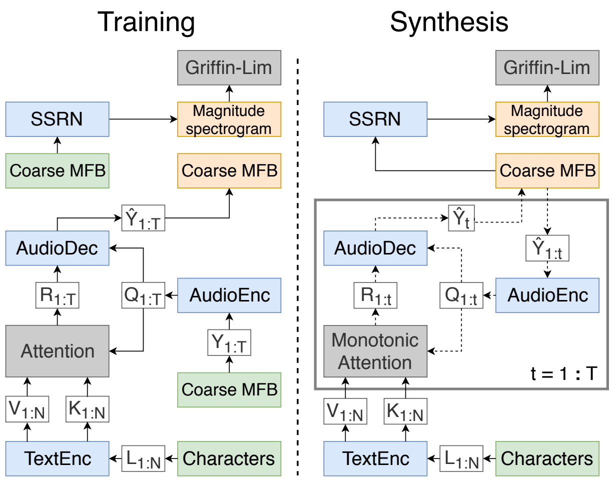

Figure 1: *DC–TTS architecture. Blue: learned modules. Grey: operations. Green: inputs. Orange: predictions. Plate notation over $t = 1 : T$ denotes the autoregressive loop at synthesis time.*

- AUDIODEC: Predicts the current coarse MFB frame $\hat{Y}_t$ (which should match the target Y_t) using $R_{0:t}$ and $Q_{0:t}$.
- SSRN: Upsamples the coarse MFB in time by a factor of 4 using transposed convolutions and reconstructs fine-grained frequency domain information.
- GRIFFIN-LIM: Finds a plausible phase spectrogram [16] for the magnitude spectrogram predicted by SSRN. After post-emphasis, an inverse FFT reconstructs the waveform.

At a low level, DC–TTS performs many 1-dimensional convolutions, these are also used to create highway layers [17] and transposed convolutions (called deconvolutions in [14]). While RNNs maintain current context using a hidden state, in DC–TTS context must be modelled either by convolution's receptive field, or by attention's summarisation of V. The modules within the autoregressive part of T2M – i.e., AUDIOENC and AUDIODEC – make use of causal convolutions, because for synthesis we cannot make use of future acoustic context. As TEXTENC and SSRN have access to future context (characters or coarse MFBs), they use non-causal convolutions.

Our model architecture closely resembles that in [14], except that: we do not use guided attention; we perform layer normalisation immediately after all (non-transposed) convolutions; and dropout rate is 0.05. Starting from 0.001, our learning rate was increased linearly for the first 4000 batches, and then decayed proportional to the inverse square of the number of batches [18, Sec 5.3], where our batch size is 16. The two models are trained using Adam [19] with an L_1 loss and binary divergence (equally weighted) on their respective predictions – coarse MFB ($\hat{Y}$) for T2M, magnitude spectrograms for SSRN.

6. Evaluation

6.1. Hypotheses

Table 1 lists our hypotheses when using training data with transcripts corrupted by the methods in Section 4.

6.2. Systems built for evaluation

To test these hypotheses we built six S2S-TTS systems, and a reference voice. Using each of the data corruption methods,

[2] Our implementation is based on `https://github.com/Kyubyong/dc_tts`

Table 1: *Hypotheses*

Clean vs corrupted data	
H_1	Systems trained on clean data are better than those trained on corrupted data.

Amount of data	
H_2	Systems trained with more data are better than those trained on less data.

Relative performance between corruption methods	
H_3	Systems trained on data corrupted by **Add 5 words** will be best because attention will learn to skip added words.
H_4	Systems trained on data corrupted by **Delete 5 words** will be worse than for **Add 5 words** because attention will not find any relevant information in the input sequence to explain certain acoustic frames.
H_5	Systems trained on data corrupted by **Replace 5 words** will be worst of all because attention will be forced to use incorrect input to explain certain acoustic frames.
H_6	Systems trained on ASR transcription output will be better than **Replace 5 words** and **Delete 5 words** because attention will find plausible (albeit not entirely correct) input to explain all acoustic frames.

we created four versions of corrupted transcripts, and for each of these we trained a separate DC-TTS system. Only 50% of the sentences were corrupted (all even-numbered ones). A fifth system was trained with the full uncorrupted dataset. A sixth system was trained on half of the sentences (all odd-numbered ones) in the full dataset. This means that all training sets included the same uncorrupted half of the full dataset, with the remaining half being either uncorrupted, corrupted, or removed. The system names and data used for training them were:

T2M$_{\text{clean-100}}$: the full uncorrupted LJ Speech data.
T2M$_{\text{clean-50}}$: half of the full data.
T2M$_{\text{ADD}}$: corrupted with **Add 5 words**
T2M$_{\text{DEL}}$: corrupted with **Delete 5 words**
T2M$_{\text{REPL}}$: corrupted with **Replace 5 words**
T2M$_{\text{ASR}}$: corrupted by ASR, which has an average WER of 34.8% for corrupted sentences.
REF: copy synthesis obtained by running ground truth coarse Mel filter banks through the SSRN model and then reconstructing waveforms with Griffin-Lim.

All systems were trained for 250 epochs, except T2M$_{\text{clean-50}}$, which was trained for 500 epochs. This ensured that the number of model updates during training was kept constant across the systems. The systems were trained using letters as input to the DC–TTS model. To synthesise the test sentences, all models used the same SSRN model, which had been trained for 500 epochs on all of the acoustic data (recall that this model is not dependent on transcriptions, so is independent of any corruptions).

6.3. Evaluation methodology: MUSHRA-like test

For each model we generated all 278 sentences from chapter 50 of the LJ Speech data set[3] which is not in the training or validation sets. We removed all sentences with acronyms because our systems have no mechanism to handle these, and all pronounced them poorly. From the remainder, we randomly selected 40 sentences for the listening test that were within a range of [-4,+4] words from the mean sentence length (across the full corpus) of 17 words.

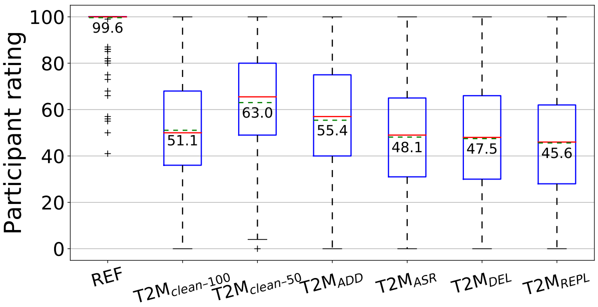

Figure 2: *MUSHRA results. Solid red lines are medians, dashed green lines are means (also given in figures), blue boxes show the 25th and 75th percentiles, and whiskers show the range of the ratings, excluding outliers which are plotted with* $+$.

We built a MUSHRA-like listening test using BeaqleJS[4], following the same framework as in [20] but without a lower bound anchor. Listeners were instructed to listen first to the reference for each screen, that this reference was hidden among the test sentences, and that they should rate the *quality* of each stimulus relative to the reference. *Quality* was intended to capture naturalness *and* intelligibility, so participants were given the correct text on each screen. Participants could not proceed to the next screen until they had listened to all the stimuli and rated at least one of them at 100. The order of the systems on a screen and the order of screens per participant were randomised. 35 native English paid participants completed the test.

7. Results

MUSHRA results are in Figure 2. The hidden reference (REF) was consistently found by listeners and correctly scored at 100.

To determine which pairs of systems are significantly different, we use Student's t-test between all 21 systems pairs with Holm-Bonferroni correction [21] to account for the large number of comparisons. All pairs were found to be significantly different at $p < 0.0005$ except for: T2M$_{\text{REPL}}$ vs. T2M$_{\text{ASR}}$ and T2M$_{\text{REPL}}$ vs. and T2M$_{\text{DEL}}$ which differ at $p < 0.05$; T2M$_{\text{ASR}}$ and T2M$_{\text{DEL}}$ are not significantly different.

The most surprising result is that T2M$_{\text{ADD}}$ significantly outperformed T2M$_{\text{clean-100}}$ (refuting H_1) – see Section 8.1.

T2M$_{\text{clean-50}}$ significantly outperformed T2M$_{\text{clean-100}}$ (refuting H_2) but we believe this is an unintended consequence of controlling the amount of training of T2M$_{\text{clean-50}}$ compared to the other systems. Since T2M$_{\text{clean-50}}$ had half the data, we trained for twice the epochs, in order to obtain the same number of weight updates. It is clear that this did not work as intended and future work should find a better training regime.

Of the models trained on corrupted transcriptions, T2M$_{\text{ADD}}$ produced the highest quality speech, followed by T2M$_{\text{DEL}}$ then T2M$_{\text{REPL}}$ (confirming H_3, H_4, and H_5).

Amongst T2M$_{\text{ASR}}$, T2M$_{\text{DEL}}$ and T2M$_{\text{REPL}}$, the differences in quality are small. So, acoustically-plausible mistranscriptions are just as harmful as arbitrary ones (refuting H_6).

[3] Speech samples are available at `https://jonojace.github.io/IS19-robustness`

[4] `https://github.com/HSU-ANT/beaqlejs`

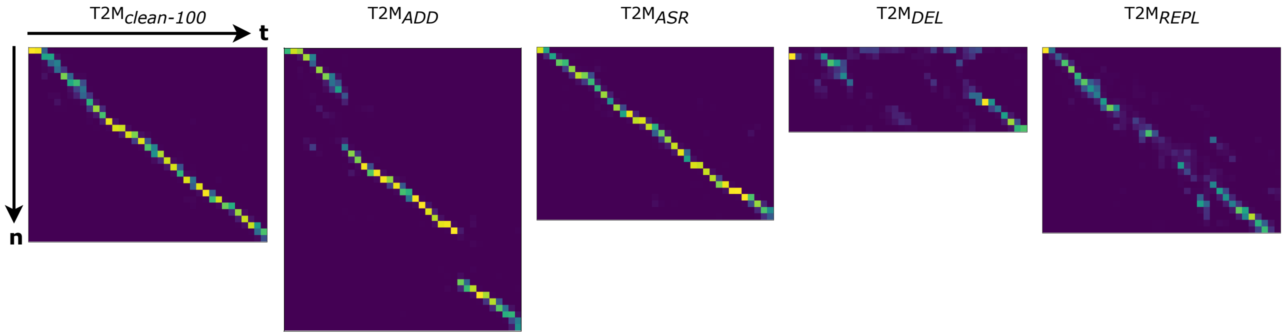

Figure 3: *Attention matrices for the training sentence with original transcription "In being comparatively modern". Encoder sequence (in characters) runs vertically and decoder output time (at 20 steps per second) runs horizontally. Matrices have a fixed number of columns because they relate to the same audio, but a varying number of rows because of the corruption to the transcription. One column depicts the attention distribution over the input characters at one decoder step. During training, T2M$_{DEL}$ and T2M$_{REPL}$ encounter some output for which there is no useful input, so they sometimes attend to the padding symbol (a zero vector). This behaviour leads to babbling: speech generation not conditioned on text.*

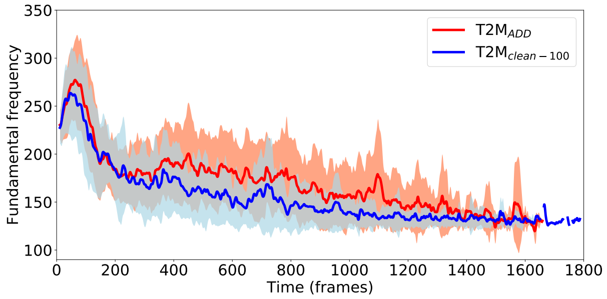

Figure 4: *Per-frame F0 mean and standard deviation for the test sentences. Blue and red lines show the mean and shaded areas show the standard deviation.*

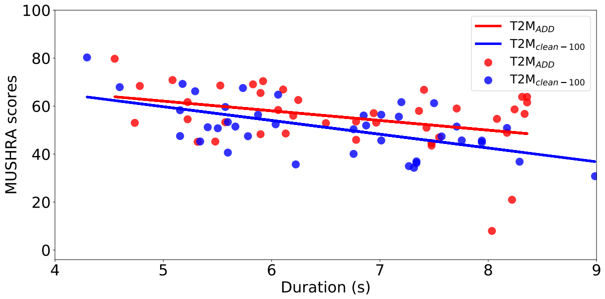

Figure 5: *MUSHRA score for each sentence (averaged across listeners) vs. sentence duration. The red and blue lines are a linear fit to the data.*

8. Analysis

8.1. Effect of sentence length

Hypothesis H_1 was refuted, which deserves further investigation. Extended listening by the authors revealed that the fundamental frequency (F0) behaviour in T2M$_{\text{clean–100}}$ was unnatural: F0 declines too quickly and does not reset, reaching a very low value and maintaining it up to the end of the sentence. Figure 4 shows how T2M$_{\text{clean–100}}$'s F0 flattens out quicker and has a narrower standard deviation than that of T2M$_{\text{ADD}}$.

Figure 5 plots quality score against sentence duration. Both systems perform poorer with longer sentences, but the trend is more pronounced for T2M$_{\text{clean–100}}$ than for T2M$_{\text{ADD}}$. The average number of letters per training sentence for T2M$_{\text{clean–100}}$ was 100, whereas for T2M$_{\text{ADD}}$ it was 117. This behaviour needs further investigation in future work.

8.2. Pronunciation errors

T2M	T2M	T2M	T2M	T2M	T2M
clean–100	clean–50	ADD	ASR	DEL	REPL
45	45	51	34	63	68

The above table reports total mispronunciations per system. The most common mispronunciations were: incorrect vowel, missing nasal, missing word-initial plosive. The clean systems and T2M$_{\text{ADD}}$ all perform reasonably well. Although T2M$_{\text{ASR}}$ was rated as low quality in the listening test, this is evidently not because of pronunciation errors since it produced fewer pronunciation errors than any other system (partially supporting H_6).

9. Conclusion

S2S models that use an attention mechanism are robust only to certain types of transcription error in the training data. Specifically, they are able to ignore *extraneous* information in the input (here, word insertions). The ability to skip over the input sequence whilst 'staying on track' in the output sequence also stems from the use of teacher forcing during training. The interaction between teacher forcing and attention needs investigation in future work. They are less capable of handling *misleading* inputs (here, word substitutions) or output that is not predictable from the input (here, word deletions).

Whilst we only explored gross word-level errors in the input, it seems reasonable to conjecture that this asymmetry applies to other kinds of mismatch between input and output sequences.

Acknowledgements: Peter Bell provided the ASR hypotheses. Zack Hodari is supported by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh. Pilar Oplustil Gallegos is funded by the Chilean National Agency of Technology and Scientific Research (CONICYT)-Becas Chile.

10. References

[1] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in *Proc. ICLR*, Toulon, France, Apr 2017.

[2] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, Stockholm, Sweden, Aug 2017, pp. 4006–4010.

[3] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: 2000-speaker neural text-to-speech," in *Proc. ICLR*, Vancouver, Canada, Apr-May 2018.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, San Diego, USA, May 2015.

[5] 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 *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, May 2019.

[6] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, and T. Drugman, "Effect of data reduction on sequence-to-sequence neural TTS," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, May 2019.

[7] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems (NIPS)*, Montréal, Canada, Dec 2018, pp. 4485–4495.

[8] A. Stan, P. Bell, J. Yamagishi, and S. King, "Lightly supervised discriminative training of grapheme models for improved sentence-level alignment of speech and text data." in *Proc. Interspeech*, Lyon, France, Aug 2013, pp. 1525–1529.

[9] N. Braunschweiler, M. J. Gales, and S. Buchholz, "Lightly supervised recognition for automatic alignment of large coherent speech recordings," in *Proc. Interspeech*, Makuhari, Chiba, Japan, Sep 2010.

[10] R. A. 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.

[11] J. Matoušek, D. Tihelka, and L. Šmídl, "On the impact of annotation errors on unit-selection speech synthesis," in *Proc. International Conference on Text, Speech and Dialogue (TSD)*, Brno, Czech Republic, Sep 2012, pp. 456–463.

[12] J. Yamagishi, Z. Ling, and S. King, "Robustness of HMM-based speech synthesis," in *Proc. Interspeech*, Brisbane, Australia, Sep 2008.

[13] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv unreviewed manuscript arXiv:1609.03499*, 2016.

[14] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, Apr 2018, pp. 4784–4788.

[15] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. ACL*, Beijing, China, Jul 2015.

[16] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[17] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in neural information processing systems (NIPS)*, Montréal, Canada, Dec 2015, pp. 2377–2385.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, Long Beach, USA, Dec 2017, pp. 5998–6008.

[19] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, USA, May 2015.

[20] I.-R. Recommendation, "1534-1,method for the subjective assessment of intermediate sound quality (MUSHRA)," *International Telecommunications Union, Geneva, Switzerland*, 2001.

[21] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, pp. 65–70, 1979.