Data Processing for Optimizing Naturalness of Vietnamese Text-to-speech System

1st Viet Lam Phung
Zalo Group, VNG Corporation
Hanoi, Vietnam
lampv@vng.com.vn

2nd Huy Kinh Phan
Zalo Group, VNG Corporation
Hanoi, Vietnam
phanhuykinh@gmail.com

3rd Anh Tuan Dinh
Zalo Group, VNG Corporation
Hanoi, Vietnam
tuanad121@gmail.com

4th Quoc Bao Nguyen
Zalo Group, VNG Corporation
Hanoi, Vietnam
baonq6@vng.com.vn

Information and Communication Technology University,
Thai Nguyen University Thai Nguyen, Vietnam

Abstract—End-to-end text-to-speech (TTS) systems have proved its great success in the presence of a large amount of high-quality training data recorded in an anechoic room with high-quality microphones. Another approach is to use available source of found data like radio broadcast news. We aim to optimize the naturalness of TTS system on the found data using a novel data processing method. The data processing method includes 1) utterance selection and 2) prosodic punctuation insertion to prepare training data which can optimize the naturalness of TTS systems. We showed that using the processing data method, an end-to-end TTS achieved a mean opinion score (MOS) of 4.1 compared to 4.3 of natural speech. We showed that the punctuation insertion contributed the most to the result. To facilitate the research and development of TTS systems, we distributed the processed data, which is known as Zalo-TTS database at https://forms.gle/6Hk5YkqgDxAaC2BU6; It consists of 18-hours of speech at a sampling rate of 44.1 kHz of one speaker with Hanoi dialect.

Index Terms—utterance selection, prosodic punctuation, end-to-end, text-to-speech, speech database

I. INTRODUCTION

Text-to-speech (TTS) systems play an important role in widely accepted, interactive systems like Siri, Microsoft Cortana, and Amazon’s Alexa. However, collecting data to build those systems is costly. Typically, a professional voice talents is recruited to read dozens of hours of text with good coverage of target domain in an anechoic room with high-quality microphone. The speakers should maintain constant fundamental frequency (F0), energy, speaking rate, and articulation throughout. There are 7000 languages in the world, and most do not receive as much research attention as English, Spanish, Mandarin, and Japanese. The so called low-resource language like Vietnamese have no carefully recorded and annotated corpora which can be used for TTS systems. The available speech corpora such as VOV (radio broadcast news) [1], VNSpeechCorpus [2], VIVOS [3] are small, and dedicated for automatic speech recognition (ASR). The VAIS-1000 [4], which is a latest database for TTS, only consists of 1000 sentences of a speaker. Another approach is to make use of various sources of found data (e.g. radio broadcast news, automatic speech recognition (ASR) corpora, and audiobooks) as TTS corpora [5]. The approach became the main challenge in the TTS evaluation of the Vietnamese Language and Speech Processing (VLSP) 2019 [6]. In the evaluation, a so-called “big training dataset”, which was recorded in different room environments by an unprofessional speaker, was provided for building TTS systems. The dataset resembles the characteristics of found data. In the paper, we proposed a data processing scheme for optimizing the naturalness of our end-to-end Vietnamese TTS systems trained on the VLSP 2019’s “found data”. Our data processing scheme consists of two key elements: 1) utterance selection using different metrics, and 2) prosodic punctuation insertion into text. In our experiments, we significantly improved the naturalness of our TTS system by applying our data processing method on training data. In the TTS evaluation of the VLSP 2019, our system achieved a MOS of 4.1 (compared to 4.3 of natural speech); which was the best result among all participants [7].

II. BACKGROUND

Researchers have attempted to build high-quality Vietnamese TTS systems in the last two decades. A text normalization method was investigated utilizing regular expressions and language model [8]. Prosodic features such as phrase breaks proved their efficacy in improving naturalness of Vietnamese TTS system [9], [10]. Different types of acoustic models were investigated such as hidden Markov model (HMM) [11], [12], and deep neural network (DNN) [13]. These HMM- and DNN-based TTS systems are limited by the oversmoothing of generated parameters [14]. A post-filtering method was proposed to compensate for the oversmoothing effect using non-negative matrix factorization [15]. Recently, the use of sequence-to-sequence model in acoustic modeling [16] in combination with neural vocoders such as WaveGlow [17]...
have enormously reduced the oversmoothing effect; thus, achieving human quality TTS [7], [18].

However, we need dozens of hours of training data for a speaker to build a high-quality TTS system. One solution is to use available sources of found data. Three types of found data: radio broadcast news, automatic speech recognition (ASR) corpora, and audiobooks were compared; showing that radio broadcast news is a good match as TTS corpora [5]. Different criteria such as standard deviation of fundamental frequency (F0), speaking rate, hypo- and hyper-articulation were explored in utterance selection for HMM-based TTS [5] and DNN-based TTS [19]. We are not aware of any attempts at utterance selection for end-to-end TTS. In this work, we explored different metrics for utterance selection such as misalignment errors, articulation, standard deviation of syllable duration, non-fluency, and standard deviation of F0.

Traditionally, an end-to-end TTS system receives a sequence of syllables, or words as input; thus, it has no explicit prosodic features incorporating in the input. The prosodic features such as phrase breaks is important for the naturalness of Vietnamese TTS system [9]. In the paper, we insert the prosodic punctuations, which corresponds to pauses in utterance, into text. We realized that the inserted prosodic punctuations led to stable, faster convergence of the training of Tacotron 2. Moreover, making use of prosodic punctuation derived from utterances is a novel way to help Tacotron 2 model learn a speaker-dependent prosodic pattern.

III. DATA

We used the “big training dataset” provided by the TTS evaluation of the VLSP 2019 [6]. There are 15000 utterances of a single speaker (approximately 23 hours) with corresponding text; which cover broadcast news. The speaker was recorded at home instead of an anechoic room. The speaker was instructed to stay in a place as quite as possible during each recording session. Therefore, the data features the three types of errors 1) variation in channel conditions, 2) mismatch between text and utterance content, and 3) variation in articulation. The variation in channel conditions are caused by microphone conditions, recording environments, channel noise, ... As a result, some utterances have mild background noise. The mismatch between text and utterance content is due to misspelling, tricky text, ... When the text was meaningless and hard to read, the speaker often gave up and said random things. The variation in articulation features hyper-articulation, inconsistency in speaking rate, F0. We downsampled the speech data from 44.1 kHz to 16 kHz.

IV. PROPOSED METHOD

In the section, we present our data processing scheme as shown in Figure 1. Given the raw utterances and corresponding raw text, we applied a noise reduction method on the audio files using minimum mean-squared-error estimator [20]. The text was normalized and tokenized [8]. The denoised audio and corresponding normalized text were aligned using an audio alignment system. Using the timestamps obtained from the alignment, we can 1) calculate different metrics of utterance selection and 2) identify and insert prosodic punctuation to text. We selected the utterances satisfying some experimental thresholds of the metrics. Each sentence was spliced into phrases by prosodic punctuation; reflecting the prosody pattern of the speaker.

A. Audio Alignment System

Vietnamese is a mono-syllabic language. We develop an audio alignment system [21]–[27] for Vietnamese to identify time-stamps from audio for syllables. We first used a voice activity detection module to segment audio into speech and non-speech segments using adaptive context attention model [28]. It is different from [19] where an HMM-based voice activity detection was used. A Time Delay Neural Network (TDNN)-based ASR system [29] was trained over-fittingly on the speech segments and corresponding normalized text in our data. We also biased a language model to the normalized text [23]. Each speech segment was decoded using the TDNN-based acoustic model and the biased language model. The resulting time-aligned transcription was aligned with the original normalized text; associating the obtained time-stamps to the original text. On the other hand, matching sequences of syllables (also called anchor points [22]) between decoding output and original text transcript can be a good indicator that the speaker read the sentence correctly. Moreover, we can use the time-stamps of pauses and silences to identify prosodic punctuation.

B. Utterance Selection Metrics

We introduce different metrics addressing three type of errors in our data:

- **Word-error rate** (WER) of decoding output when comparing to original text addresses the mismatch between text and utterance content. Every utterances with a WER less than 90% were removed from our data; thus, we removed 800 utterances.

- **Articulation** [5] is used to address the variation in articulation or abnormal articulation. It is calculated as in Equation 1 where $P_{signal}$ is the power of speech segments extracted from voice activity detection module; the average syllable duration (avg.syl.dur) is calculated based...
on the aligned time-stamps (obtained in IV-A). Speech segment with high articulation is hyper articulated. The hyper-articulated speech is unnatural because it has slow speaking rate and high energy [30].

\[
\text{Articulation} = P_{\text{signal}} \times \text{avg.syl.dur} \tag{1}
\]

- **Standard deviation of syllable duration** \(\text{std.syl.dur}\) is used to address the inconsistency of speaking rate. The duration of each syllable is calculated according to the aligned time-stamps. Given a speech segment, a high value of \text{std.syl.dur}\) indicates that the narrator spoke sometimes fast and sometimes slow within the segment. Speech segments with high inconsistency of speaking rate are unnatural.
- **Non-fluency** is used to address the reading non-fluency or variation in articulation. Moreover, the alignment procedure in IV-A can have misalignment errors. We can also use the non-fluency metric to address the misalignment errors. The non-fluency is calculated as in Equation 2 where maximum duration of internal silence and average syllable duration are calculated according to aligned time-stamps. The internal silence is silence or pause other than the start and end ones. A high value of non-fluency indicates a long pause within an utterance; reflecting the non-fluency.

\[
\text{Non-fluency} = \frac{\text{max (internal silence duration)}}{\text{avg.syl.dur}} \tag{2}
\]

- **Standard deviation of F0** \(\text{std.F0}\) is used to address inconsistency of F0. A high value of \text{std.F0}\) can be due to more expressive speech. We removed utterances with high values of \text{std.F0}\).

For each metric, we rejected the 5% of data corresponding to the segments with the worst values of the metric.

**C. Prosodic Punctuation Insertion**

We detected four types of prosodic punctuation from speech based on the duration of internal silences. The internal silence duration can be calculated using aligned time-stamps. We represent each type of prosodic punctuation with a special character. We then insert the special characters into text at the positions of corresponding silences. By experiments, we determined four ranges of silence durations to identify the prosodic punctuations: \([0.12, 0.15], (0.15, 0.21], (0.21, 0.27]\) and more than 0.27 second. The prosodic punctuations are also used to mark the bad pauses caused by non-fluency. Thus, we can prevent the models to align the bad pause frames to any syllables.

**V. VIETNAMESE TTS SYSTEM**

Our end-to-end TTS system have two components: 1) a encoder-decoder acoustic model and 2) a neural vocoder. The encoder-decoder acoustic model converts a sequence of syllables with prosodic punctuations to a 80-dimensional Mel-spectrogram. The neural vocoder generates speech from the 80-dimensional Mel-spectrogram. In normalized text, we consider tokens, inserted prosodic punctuation as syllables. In the paper, we utilized Tacotron 2 [16] for acoustic modeling, and WaveGlow vocoder [17]. As a result, our end-to-end system can achieve a real-time inference speed.

**A. Encoder-decoder acoustic model**

Generally, the acoustic model of an E2E SPSS system has a encoder-decoder structure [31] equipped with an attention mechanism [32]. The encoder-decoder structure maps a source sequence \(x_{1:n} = (x_1, \ldots, x_n)\) to a target sequence \(y_{1:m} = (y_1, \ldots, y_m)\) which can have different lengths, i.e. \(n \neq m\). In our case, the source sequence \(x_{1:n}\) is the sequence of syllables represented as one-hot vectors. The target sequence \(y_{1:m}\) is the sequence of 80-dimensional mel-spectrum. First of all, the encoder (Enc) maps the source sequence \(x_{1:n}\) to a sequence of hidden representations \(h_{1:n} = (h_1, \ldots, h_n)\). The decoding of the target sequence is autoregressive, which means the previously generated vectors are considered as addition input at each decoding step \(t\). To generate an output vector \(y_t\), a weighted sum of \(h_{1:n}\) forms a context vector \(c_t\), where the weight vector is a calculated attention probability vector \(a_t = (a_t^{(1)}, \ldots, a_t^{(n)})\). We can think about the attention probability \(a_t^{(k)}\) as the importance of hidden representation \(h_k\) at time \(t\). Finally, the decoder (Dec) use the context vector \(c_t\) and the previously generate features \(y_{1:t−1}\) to decode \(y_t\). Note that the attention calculation and decoding process take the previous hidden state of the decoder \(q_{t−1}\) as an input. We can formulate the above procedure as follows:

\[
\text{h}_{1:n} = \text{Enc}(x_{1:n}) \tag{3}
\]

\[
a_t = \text{attention}(q_{t−1}, h_{1:n}) \tag{4}
\]

\[
c_t = \sum_{k=1}^{n} a_t^{(k)} h_k \tag{5}
\]

\[
y_t, q_t = \text{Dec}(y_{1:t−1}, q_{t−1}, c_t) \tag{6}
\]

**B. WaveGlow vocoder**

We used WaveGlow [17] as for neural vocoding. WaveGlow is a deep generative model for audio that integrates Glow, a generative model for image processing, [33] with WaveNet [34]. During training, a speech waveform \(o\) is converted to a Gaussian white noise \(z\). Conversely, a Gaussian white noise is converted to a speech waveform by the inverse operation during inference process. By introducing the invertible \(1 \times 1\) convolution and affine coupling layers, the loss function of the WaveGlow vocoder is calculated as in Equation 7.

\[
- \log p_\theta(o) = \frac{z(o^T)z(o)}{2\sigma_{WG}^2} - \sum_{j=0}^{m} \log s_j(y, f) - \sum_{k=0}^{n} \log|\det(W_k)|, \tag{7}
\]

The \(\theta\) denotes network parameters; \(f\) is conditional acoustic features. The \(s_j, W_k,\) and \(\sigma_{WG}^2\) are output coefficients of
VI. EVALUATION

In the section, we trained an end-to-end TTS system on provided original data (or “big training dataset”) as our Baseline system. We then evaluated the efficacy of our data processing scheme by comparing systems trained on the processed data to the baseline system. The US denotes that the original data was processed by utterance selection alone. The Punc denotes that only prosodic punctuation was used on original data. The Punc&US denotes that both utterance selection and prosodic punctuation insertion were used. In total, we trained four end-to-end TTS systems. We leave-out 32 sentences for testing. With each case of data processing, we used 90% or remaining sentences for training and the other 10% validation. Only training and validation text has prosodic punctuations but not testing text. We used an FFT-length of 1024, a hop size of 256 and a window size of 1024. We used a 80-channel mel-filterbank, spanning from 95 Hz to 7600 Hz, followed by log dynamic range compression. We trained Tacotron 2 up to 50,000 iterations using an Adam optimizer with a learning rate of $10^{-3}$ exponentially decaying to $10^{-4}$ from the 30,000th iteration. The WaveGlow vocoder was trained up to 350,000 iterations using weight normalization [35] and an Adam optimizer with a fixed learning rate of $10^{-4}$. We used a batch size of 32 and 8 for Tacotron 2 and WaveGlow, respectively. We trained the networks on a Nvidia GeForce RTX 2080 Ti. We submitted the Punc&US system to the VLSP 2019’s evaluation [6], [7]. The Nat denotes target natural speech.

A. VLSP 2019’s Evaluation

A MOS test was conducted to compare our submitted system to other participants in the evaluation. There are more than 30 participating groups from academy and industry in the evaluation. There are 24 test sentences included by the organizer of VLSP. We were not able to do prosodic punctuation insertion because test audio files were not available. The synthesized utterances were presented to three groups of listeners: speech experts, and undergraduates. There were 28 Vietnamese (10 experts + 18 students) listeners. At each trial, a listener was asked to rate the quality of a utterance in a 5-point scale: “definitely better” (+2), “better” (+1), “same” (0), “worse” (−1), “definitely worse” (−2). The test involved 32 sentences, and 10 system pairs; resulting in 32 × 10 = 320 unique trials. The perceived loudness differences between these stimuli were minimized using a root-mean-square A-weighted (RMSA) measure [36]. We limited each listener to hear each unique sentence once (presentation order was randomized); therefore we need 320 ÷ 32 = 10 listener to cover all trials. We recruited 20 participants who are native Vietnamese speakers. Table I shows the pair-wise relative quality of the systems.

To approximate the ordering between all systems, we projected the non-negative pair-wise relative quality matrix to a single dimension using multiple dimensional scaling (MDS). Figure 2 shows the results. All data processing methods can improve the quality of TTS system. The results suggested that using our proposed method is efficient in optimizing the naturalness of end-to-end TTS system. By using our data processing method (Punc&US), which includes both utterance selection and punctuation insertion, we achieved close quality to natural speech (NAT). Interestingly, the prosodic punctuation insertion (Punc) is more efficient than utterance selection (US) in improving the quality of baseline TTS system. The reason is the dysfluency of the speakers, which introduced unnecessary pauses or stops to every utterances. Without marking the pauses, an E2E SPSS systems will treat them as parts of the pauses, an E2E SPSS systems will treat them as parts of

\[ j \text{th WaveNet in the affine coupling layers, the } k \text{th weighting matrix of the invertible } 1 \times 1 \text{ convolution layer, and the assumed variance of the Gaussian distribution, respectively.} \]
words, making the alignment between text and speech during training unstable and incorrect. These results suggest that if we prioritize the non-fluency metric in utterance selection, we can improve the performance of utterance selection. Therefore, when applying utterance selection for E2E SPSS system, we should not treat all metrics equally as applying for HMM- or DNN-based TTS [5], [19].

VII. CONCLUSION

In this paper, we proposed a data processing technique including utterance selection and prosodic punctuation insertion. We showed that using the data processing method can improve the quality of end-to-end TTS system trained on found data. In a VLSP 2019’s evaluation, our system achieved a MOS result of 4.1, which was the best MOS result among participants in the evaluation. Our CMOS test showed that the punctuation insertion contributed more to the result than the utterance selection. All of the processing methods can improve the quality of end-to-end TTS system trained on found data. In future works, we will predict the prosodic punctuation for test sentence from text. We will explore the importance of each individual metric to improve the performance of utterance selection. We distributed the so-called Zalo-TTS database at https://forms.gle/6Hk5YkgDxAaC2BU6; It consists of 18-hours of speech data at a sampling rate of 44.1 kHz of one speaker with Hanoi dialect.

REFERENCES

[1] L. C. Mai and N. D. Dung, “Design of vietnamese speech corpus and current status,” in Proceeding of ICSLP, San Diego, Kent Rigle, Singapore, 2006.

[2] V.-B. Le, D.-D. Tran, E. Castelli, L. Besacier, and J.-F. Serignat, “Spoken and written language resources for vietnamese,” in Proceedings of LREC, Lisbon, Portugal, 2004.

[3] H.-T. Luong and H.-Q. Vu, “A non-expert kaldi recipe for vietnamese speech recognition system,” in Proceedings WSLI-1 & OIAF4HLT-2, Osaka, Japan, 2016.

[4] Q. T. Do and L. C. Mai, “Vais-1000: a vietnamese speech synthesis corpus,” in IEEE Dataport, 2017.

[5] E. Cooper, “Text-to-speech synthesis using found data for low-resource languages,” Ph.D. dissertation, Columbia University, 2019.

[6] “Vietnamese language and speech processing 2019 summary,” https://vlspl.org.vn/sites/default/files/2019-10/VLSP2019-TTS-summary.pdf.

[7] P. V. Lam, P. H. Kinh, D. A. Tuan, T. K. Duy, and N. Q. Bao, “Development of zalo vietnamese text-to-speech for vlp 2019,” http://vlspl.org.vn/sites/default/files/2019-10/VLSP2019-TTS-Vietnam.pdf, accessed: Oct, 2019.

[8] D. A. Tuan, P. T. Lam, and P. D. Hung, “A study of text normalization in vietnamese for text-to-speech system,” in Proceedings of Oriental COCOSDA Conference, Macau, China, 2012.

[9] A. T. Dinh, T. S. Phan, T. T. Vu, and C. M. Luong, “Vietnamese hmm-based speech synthesis with prosody information,” in Eighth ISCA Workshop on Speech Synthesis, Barcelona, Spain, 2013.

[10] T. T. T. Nguyen, “Hnn-based vietnamese text-to-speech : Prosodic phrasing modeling, corpus design system design, and evaluation,” Ph.D. dissertation, Laboratoire d’Informatique pour la Mécanique et les Sciences de l’Ingénieur (UPR 3251), 2015.

[11] T. S. Phan, A. T. Dinh, T. T. Vu, and C. M. Luong, “An improvement of prosodic characteristics in vietnamese text to speech system,” in Knowledge and Systems Engineering, V. N. Huyhn, T. Denoux, D. H. Tran, A. C. Le, and S. B. Pham, Eds. Cham: Springer International Publishing, 2014, pp. 99–111.

[12] D. K. Ninh, “A speaker-adaptive hmm-based vietnamese text-to-speech system,” in 2019 41st International Conference on Knowledge and Systems Engineering (KSE), Oct 2019, pp. 1–5.

[13] T. V. Nguyen, B. Q. Nguyen, K. H. Phan, and H. V. Do, “Development of vietnamese speech synthesis system using deep neural networks,” in Journal of Computer Science and Cybernetics, vol. 34, no. 4, 2019, pp. 349–363.

[14] T. Toda, A. W. Black, and K. Tokuda, “Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter,” in ICASSP, 2005.

[15] A.-T. Dinh, T.-S. Phan, and M. Akagi, “Quality improvement of vietnamese hmm-based speech synthesis system based on decomposition of naturalness and intelligibility using non-negative matrix factorization,” in International Conference on Advances in Information and Communication Technology, 2016, pp. 490–499.

[16] 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. Yu, “Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions,” CoRR, vol. abs/1712.05884, 2017. [Online]. Available: http://arxiv.org/abs/1712.05884

[17] R. Prenger, R. Valle, and B. Catanzaro, “Waveglow: A flow-based generative network for speech synthesis,” in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2019, pp. 3617–3621.

[18] 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

[19] F. Kuo, S. Aryal, G. Degottex, S. Kang, P. Lanchantin, and I. Ouyang, “Data selection for improving naturalness of tts voices trained on small found corpuses,” in 2018 IEEE Spoken Language Technology Workshop (SLT), Dec 2018, pp. 319–324.

[20] D. Yu, L. Deng, J. Dropcho, J. Wu, Y. Gong, and A. Acero, “A minimum-mean-square-error noise reduction algorithm on mel-frequency cepstra for robust speech recognition,” in ICASSP, 2008, pp. 4041–4044.

[21] J. Roberty-Ribes and R. Mukhtar, “Automatic generation of hyperlinks between audio and transcript,” in Eurospeech, 1997.

[22] P. J. Moreno, C. Joerg, J.-M. V. Thong, and O. Glickman, “A recursive algorithm for the forced alignment of very long audio segments,” in International Conference on Spoken Language Processing, 8, Ed., 1998.

[23] L. Lamel, J. Gauvain, and G. Adda, “Lightly supervised and unsupervised acoustic model training,” Computer Speech and Language, vol. 16, pp. 115–129, 2002.

[24] A. Venkataraman, A. Stolcke, W. Wang, D. Vergyri, V. Gadde, and J. Zheng, “An efficient repair procedure for quick transcriptions,” in ICSLP, 2004.

[25] H. Chan and P. Woodland, “Improving broadcast news transcription by lightly supervised discrimination training,” in ICASSP, 2004, pp. 737–740.

[26] O. Boeffard, L. Charonnat, S. L. Maguer, D. LoliVe, and G. Vidal, “Towards fully automatic annotation of audiobooks for tts,” in International Conference on Language Resources and Evaluation, 2012.

[27] P. Lanchantin, P. Karanasou, M. J. F. Gales, X. Liu, L. Wang, Y. Qian, and C. Zhang, “The design of the Cambridge multi-genera broadcast challenge systems for the multi-genre broadcast challenge,” in ASRU, 2015.

[28] J. Taekim and M. Hahn, “Voice activity detection using an adaptive context attention model,” EEE Signal Processing Letters, vol. 25, no. 8, pp. 1181–1185, 2018.

[29] V. Peddinti, D. Povey, and S. Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” in Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[30] J. Hirschberg, D. Litman, and M. Swerts, “Prosodic and other cues to speech recognition failures,” Speech Communication, vol. 43, pp. 155–175, 2004.

[31] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, ser. NIPS’14. Cambridge, MA, USA: MIT Press, 2014, p. s104–s311.

[32] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by conditioning wavenet on mel spectrogram predictions,” [Online]. Available: http://arxiv.org/abs/1609.04978

[33] D. P. Kingma and D. P. W. H. P. Background, “Glow: Generative flow with invertible 1 x 1 convolutions,” NeurIPS, pp. 10 215–10 224, 2018.
[34] 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

[35] T. Salimans and D. P. Kingma, “Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks,” arXiv e-prints, p. arXiv:1602.07868, Feb. 2016.

[36] International Electrotechnical Commission, Electroacoustics-sound level meters-part 1: Specifications, 61672, 2002.