

Interactive Text-to-Speech System via Joint Style Analysis

Yang Gao^{1,*}, Weiyi Zheng², Zhaojun Yang², Thilo Köhler², Christian Fuegen², Qing He^{2,†}

¹Carnegie Mellon University

²Facebook AI

yanggao@andrew.cmu.edu, {wyz, zhaojuny, tkoehler, fuegen, qinghe}@fb.com

Abstract

While modern TTS technologies have made significant advancements in audio quality, there is still a lack of behavior naturalness compared to conversing with people. We propose a style-embedded TTS system that generates styled responses based on the speech query style. To achieve this, the system includes a style extraction model that extracts a style embedding from the speech query, which is then used by the TTS to produce a matching response. We faced two main challenges: 1) only a small portion of the TTS training dataset has style labels, which is needed to train a multi-style TTS that respects different style embeddings during inference. 2) The TTS system and the style extraction model have disjoint training datasets. We need consistent style labels across these two datasets so that the TTS can learn to respect the labels produced by the style extraction model during inference. To solve these, we adopted a semi-supervised approach that uses the style extraction model to create style labels for the TTS dataset and applied transfer learning to learn the style embedding jointly. Our experiment results show user preference for the styled TTS responses and demonstrate the style-embedded TTS system’s capability of mimicking the speech query style.

Index Terms: Text-to-speech synthesis, emotion, style, semi-supervised

1. Introduction

With increasing interest in interactive speech systems such as voice assistants, there is an increased demand for human-like text-to-speech (TTS) systems. While recent technology advancements in speech synthesis have achieved human-like audio quality [1–4], the TTS’s speaking style does not mimic the naturalness and expressiveness as in human conversations, because conventional speech interfaces respond to input speech queries with default speaking style learned from the TTS training dataset. To make the TTS more interactive, the TTS’s response should vary depending on the context and the speaking style of the input speech query. For example, when the user is speaking fast and rushing out the door in the morning, the TTS would match the hurried pace; and when the user is in a quiet place and is speaking softly, the TTS would respond with a soft and quiet voice. By detecting the input speech query’s style and generating response accordingly, TTS can provide a more natural and customized user experience. One way to achieve this interaction is to incorporate two key components: a style extraction model that detects the speaking style of the input speech query and generates a style embedding, and a multi-style TTS system that can synthesize styled speech with respect to different style embedding inputs.

The challenge lies in jointly training the style extraction model and the multi-style TTS so that the style embeddings generated by the style extraction model can be genuinely respected by the TTS, even though the two components are trained with different datasets and style labels. In this paper, the TTS dataset is a commissioned dataset recorded with professional voice talents. Only a small part of the TTS dataset has style labels. For the style extraction model, we make use of the external IEMOCAP dataset. These two datasets have different style labels. In order to achieve consistent labels between TTS training data and unseen queries, we incorporated both IEMOCAP dataset and a small portion of the labeled TTS dataset in the style classifier model’s training.

We first train a multi-modal style classifier using the IEMOCAP dataset with the model described in [5]. Taking the softmax layer of the style classifier as style embedding, the classifier serves as a style embedding extraction model. This model is applied to the unlabelled TTS dataset to generate the style embeddings in a semi-supervised fashion. By using the style embedding as additional auxiliary features for the TTS system, we could train a controllable multi-style TTS system that learns to respect given target styles. During speech synthesis, style embedding is first extracted from the input speech query and then fed into the TTS system to produce response in matching styles. In summary, we developed an interactive multi-style TTS system that could lead to natural, expressive human-machine speech interactions. The multi-style TTS system is evaluated using comprehensive subjective experiments.

2. Related work

2.1. Emotion recognition

Early approaches on emotion recognition have mostly been inspired by psychology studies [6, 7]. Recently, deep neural networks (DNNs) have first been used to learn high-level representations for utterance-level emotion recognition [8]. Trigeorgis *et al.* further applied convolutional neural networks (CNNs) to model context-aware emotion-relevant features, which are then combined with long short-term memory (LSTM) networks aiming towards end-to-end emotion modeling [9]. Fundamentally, the expression of emotions is usually conveyed through multi-modal behavior channels, including speech, language, body gestures, or facial expressions. Thus, emotion recognition is often formulated as a classification problem of utterances using these multi-modal signals. Reference [10] proposed a multi-modal dual recurrent encoder to simultaneously model the dynamics of both text and audio signals within an utterance to predict emotion classes. This architecture has achieved state-of-the-art performance on IEMOCAP [11] dataset, which is a multi-modal emotion dataset and has been widely used in the affective computing community.

* Work performed during internship at Facebook AI

† Corresponding author

2.2. Expressive TTS

One popular topic in the recent research of TTS is expressive TTS. Expressive TTS has been studied for years from the HMM-based synthesis using style modeling with control vector [12–14] to the state-of-the-art prosody transfer expressive TTS work [15–17], which is aiming at achieving controllable style synthesis in TTS. However, to learn and synthesize specific styles, there are limitations with unsupervised style factorization learning [17]. Since the disentanglement of different styles is heavily influenced by randomness and the choice of hyper-parameters [18], the learning of specific target styles is not completely controllable.

Under supervision with explicit prosody labels, the styles could be learned with direct guidance [19–21]. Supervised learning requires a large amount of labeled data, giving difficulties in the development of expressive TTS research and applications. Furthermore, the data labels for styles may not be well overlapped with the needs. An approach to tackle this is proposed in [5]. But, the external dataset and the synthesis dataset Blizzard 2017 [22] have differences in background noise, recording environment, speech quality, etc. With the differences between these two datasets, the classifier trained using an external dataset may not be well-adapted to extract representations from the synthesis data. The final emotion synthesis accuracy is 41% on four emotions [5] evaluated by listeners, which may be caused by the domain gap between the TTS dataset and the external dataset.

3. Datasets

3.1. TTS dataset

The TTS dataset was recorded in voice production studio by multiple professional voice talents with 24kHz sampling rate. It has balanced phonemic and textual information. After labelling to accommodate with the task, 7% of the TTS dataset has utterance level style labels including happy, sad, neutral, angry, fast, and soft. Details of the data are summarized in Table 1. These utterances are used, as additional data, to train a multi-speaker style classifier, described in Section 4.1. To train the multi-style TTS, we use 40,244 utterances from a single speaker which contains around 3000 style labelled utterances. The style embeddings for unlabelled portion are extracted using the style extraction model, more details in Section 4.1.

3.2. IEMOCAP dataset

To compensate for the limited amount of labeled data in our TTS dataset, we chose IEMOCAP [11], which is widely used for emotion recognition, to complement our training data. In this dataset, both video and audio were recorded from ten actors in dyadic sessions under scripted and spontaneous communication scenarios. The dataset contains 12.5 hours of recordings with a sampling rate of 22kHz. Each utterance contains one emotion label, such as neutral, happy, sad, anger, surprise, etc. To be consistent with former research [5,10] and also be suitable for our own interaction goal, we select the following emotions in our study: neutral, happy, sad, and angry. Similar to the approach in [10], we merge utterances with excited emotion with those of happy emotion.

Table 1: Training data style label statistics

Dataset	Split	Fast	Soft	Neutral	Happy	Angry	Sad
TTS Dataset	Train	1145	1814	4481	885	140	35
	Dev	105	161	439	79	13	3
	Test	124	220	506	93	17	2
	All	1374	2195	5426	1057	170	40
IEMOCAP	Train	–	–	1390	1307	865	883
	Dev	–	–	100	90	61	62
	Test	–	–	218	239	177	139
	All	–	–	1708	1636	1103	1084

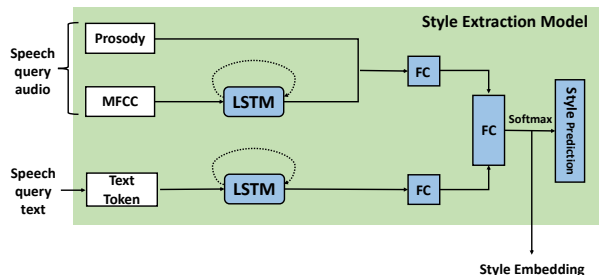


Figure 1: Multimodal style extraction model

4. Framework and Models

4.1. Semi-supervised style transfer learning

For speech style classification, we used the multimodal dual recurrent encoder (MDRE) model adapted from [10]. As shown in Figure 1, the model is composed of two separate recurrent encoders for audio and text modeling, respectively. The audio model uses 39 dimensions Mel-frequency Cepstral Coefficients (MFCC) features and utterance level prosody feature extracted using openSMILE [23] as inputs, and the text model uses 300-dimension embeddings to represent each word token. The MFCC, prosody, and text features are the same as described in [10]. The audio encoder output is concatenated with the text encoder output, then fed into a fully-connected layer to produce the final classification. We changed the loss function from sigmoid cross-entropy to softmax cross-entropy as it produced significantly better results for our training task. We use the softmax layer output as embedding features, which can be interpreted as a weighted representation of different speaking styles. The softmax feature as embedding is shown in Figure 1.

The style classifier is used to generate style embedding from the speech query during inference, as well as to extract style embedding for the TTS training dataset. At first, we trained the style classifier using the IEMOCAP dataset and applied it to generate style features on the TTS dataset. However, the classifier gives inaccurate predictions on the TTS dataset due to domain mismatch between the TTS dataset and the IEMOCAP dataset. Therefore, we labeled a small part of our TTS dataset using one label per utterance as Table 1 and fine-tuned the style classifier using these labels as in Section 5.2.

4.2. Multi-style TTS system

Figure 2 shows the architecture of the expressive TTS system. It consists of a style embedding extraction component that generates the style embedding from speech query and a multi-style TTS, which uses the style embedding to synthesize its response in matching style. As shown in Figure 2, our TTS pipeline is a multi-model framework that consists of a linguistic frontend, a prosody model, an acoustic model, and a conditional neural vocoder. Specifically, the input text is first converted to linguis-

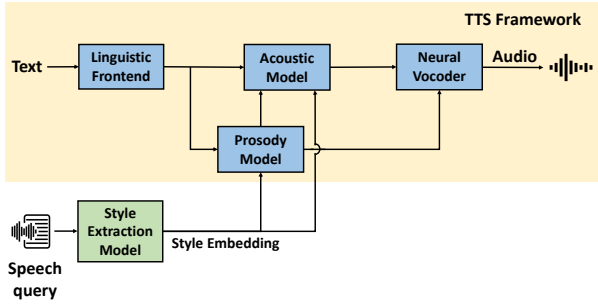


Figure 2: *Style embedded TTS framework: the style extraction model generates the style embedding based on the user speech query (text + audio), which is used to condition the TTS synthesis*

tic features. Then, the linguistic features, along with any conditional features such as style embedding, speaker IDs are used to produce the prosodic features such as duration and F_0 . The prosody model consists of a single layer LSTM model with 256 hidden units with content-based global attention [24], whose context vector contains linguistic features of the entire utterance. It is important to build a separate prosody model in the pipeline because it allows easier control for the speech style during synthesis time. Then, linguistic features combined with prosodic features are used to generate the 13-dim MFCC spectral acoustic features. The acoustic models consist of a two layer uni-directional LSTM with 256 hidden units per layer. At the last stage, a conditional neural vocoder using the WaveRNN [25], takes in the 13-dim MFCC along with the F_0 feature to synthesize a 24kHz audio waveform. Our WaveRNN model consists of a single layer gated recurrent unit (GRU) with 1024 hidden units. The speaking style of the synthesized speech is controlled by the conditional style embedding feature, which can be pre-defined or extracted using the style extraction model from the input query, as in Figure 2.

5. Experiments and results

5.1. Implementation details

The style classification model is adapted from [10] and is shown in Figure 1. Specifically, we set the batch normalization layer with 0.9 momentum to help cross-domain adaptation. To compensate for the imbalance among style labels, we weighted the by-class loss function and the per-class accuracy with an inverse of style label prior and capping the neutral label prior to 0.25. Besides, AdaBN [26] is implemented in this model to boost domain adaptation performance between the TTS and multi-style datasets.

The multi-style TTS system is trained using the commissioned TTS dataset with style embedding features as conditional input features. The style embedding labels were generated by passing each utterance through the style classification model, as described in Section 4.1. In the synthesis phase, the style embedding features could be automatically extracted from the input query or manually assigned as a combination of different styles.

5.2. Style classification

In the style classification task, we first tested the style classifier model performance on the IEMOCAP train/test split. It achieves an overall accuracy of 72.7%, which is similar to

the reported state-of-the-art [10]. To improve the embedding quality on the TTS dataset, the IEMOCAP dataset and the labeled subset of the TTS dataset were combined during training. The results show that the style classifier achieves 91.4% overall accuracy and 71.5% weighted accuracy on the TTS labeled dataset. With a lack of labeled data in anger and sadness in the TTS dataset, the prediction accuracy of these two classes is not high. The style classification accuracy decreased slightly on the IEMOCAP dataset after joint training, likely due to the mismatch between the TTS and IEMOCAP datasets.

We performed normalization on the input features. The normalization is performed corpus-wise to compensate for the domain difference between our TTS dataset and the IEMOCAP dataset. Table 4 shows that normalizing both MFCC and prosody provides the best classification accuracy on the TTS dataset’s validation set. So in the final model, we normalized both MFCC features and prosody features. The final classification accuracy for the TTS dataset is in Table 2.

5.3. Multi-style TTS with conditional style embedding

To evaluate our multi-style TTS’s performance, we collected subjective evaluation responses from 22 listeners. As reported in [5, 27, 28], the human perception on the emotions of natural speech is only around 50%, showing the ambiguity of emotion perception. Hence, instead of evaluating the subjective style accuracy on the multi-style synthesis results, we conducted the ABX test and preference test. Synthesis samples of our system are available at [29].

5.3.1. ABX test

The ABX test is designed to evaluate whether two styles generated with the same style embedding are perceived to be closer in speaking style when compared to a sample with a different style embedding. Since the style embedding can be used as a probability distribution over the 6 styles, to synthesize audio in a certain style, we construct the style embedding vector to have a value of 0.95 for the selected style and 0.01 for the other five styles. We designed the ABX test as follows. Given two different styles, we randomly choose an example in each style. We denote these two examples as A and B . We then randomly choose a different sample X from one of these two styles as reference. We then ask the listener to listen to samples A , B , and X , and then select which of A or B is perceived to be of the same style as the reference X .

We created 15 test sets in total, each of which corresponds to a pair of styles A and B , and a reference X . 22 listeners participated in the test, which gives a total of 330 ABX test comparison scores. We achieved an overall accuracy of 82.42% (i.e. total number of matching pairs divided by the total number of ABX tests), indicating that the multi-style TTS is able to generate samples with perceivably distinguishable styles.

5.3.2. Preference test

The preference test is designed to compare TTS responses generated by a default TTS without multi-style capability and the multi-style TTS when the style embedding is explicitly provided. Specifically, we ask the listeners to choose between TTS responses synthesized with the same text but different models: baseline TTS model (i.e., TTS without style embedding) or the multi-style TTS model. For the multi-style TTS responses, we provide the utterance with style of either the neutral style or, when appropriate, a hand-crafted style embedding (i.e., other

Table 2: Style classification on TTS data: AdaBN helps the domain adaptation between IEMOCAP dataset and TTS dataset, improving the weighted accuracy of six style classes.

Dataset	Trick	Neutral	Fast	Soft	Happy	Angry	Sad	Accuracy	
								Weighted	Unweighted
Train	BN	0.984	0.871	0.964	0.892	0.176	0.0	0.779	0.973
	AdaBN	0.953	0.847	0.918	0.903	0.353	0.0	0.915	0.957
Dev	BN	0.979	0.819	0.994	0.81	0.385	0.0	0.686	0.931
	AdaBN	0.927	0.8	0.963	0.873	0.538	0.333	0.766	0.904
Test	BN	0.984	0.871	0.964	0.892	0.176	0.0	0.683	0.940
	AdaBN	0.953	0.847	0.918	0.903	0.353	0.0	0.715	0.914

Table 3: TTS data F0 statistics: the Happy style has higher mean F0 than other styles. And the F0 standard deviations of Angry, Happy and Sad are larger than Neutral, Fast and Soft styles.

Style	Angry	Happy	Sad	Neutral	Fast	Soft
F0	195.5±30.8	214.8±37.3	197.3±30.8	183.7±10.3	181.9±12.8	180.5±14.7

Table 4: Feature selection: Normalizing MFCC and prosody vector can improve the performance of style classifier.

Features	Accuracy	
	Weighted	Unweighted
Unnormalized	0.726	0.875
Normalized MFCC	0.673	0.840
Normalized prosody	0.494	0.62
Normalized both	0.766	0.904

Table 5: Subjective Preferences: the proposed TTS model’s results are preferred over the baseline TTS model’s.

	Baseline TTS	Multi-style TTS	
		Neutral Style	Other Styles
Preference (%)	28.0	54.2	17.8

style) assigned as a soft label whose style weights are determined based on the content of the utterance.

Results in Table 5 show that the multi-style TTS is preferred over the baseline TTS 72% of the time, indicating strong user-preference when an appropriately styled TTS response is provided. It is interesting to note that the neutral style from the multi-style TTS is preferred by the listeners most of the time. This is largely due to the content of the test utterances, which is best spoken with a peaceful and relaxing neutral style. This result is consistent with the findings in [5], which states that listeners prefer appropriate variation over random variation.

5.3.3. Mimicking real life input query with styled TTS response

We conducted experiments to evaluate the generalization capacity of the close-loop style extraction and multi-style TTS system. We recorded speech queries from multiple speakers who have never been seen in the training of our framework. These speakers read the queries freely in a quiet conference room. We then generated TTS responses for each query by conditioning on its style embedding. Our results show that over 40% of test pairs are evaluated as good matches by listeners. We noticed that when the speaking style of the input query is strong, the TTS response can match the input style to a certain extent (samples are at [29]). This can potentially be improved with

more coherent style labels between the style extraction model training data and the TTS dataset.

6. Discussions

In our proposed system, the soft style embedding is a weighted representation of different styles such that increasing the weight of a certain style emphasizes that style’s effect on the synthesis outputs, shown in [29]. This demonstrates the multi-style TTS’s capability of synthesizing styled-speech with respect to the soft style embedding. We noticed utterance-mean F0 differs for different styles in the synthesis results, representing the style difference. For example, the inference result of the Happy style has a significantly higher F0 mean than the other styles. This is consistent with the statistics of F0 for different predicted classes in TTS training data, as shown in Table 3.

We also noticed that the the Happy style of multi-style TTS has a significantly higher F0 mean than the other styles. This could be due to the reason that the model focused on the most distinguishable feature, such as F0 mean and failed to learn the nuances of the F0 contour. To mitigate this problem, the F0 mean and the F0 contour can be modeled separately. In addition, the sad and angry styled audio quality was comparatively worse than other styles, which could be due to the lack of anger and sadness samples in the TTS dataset. In the future, the performance of the multi-style TTS system can be further improved with a training dataset that contains more balanced style labels.

7. Conclusions

In conclusion, we attempted to develop a style-embedded TTS that is more contextual and interactive. As shown in Section 5.3, with perfect style embedding, the system generated preferred TTS responses compared to a single style TTS. With automatically extracted style embeddings from real speech queries, the system demonstrated moderate capability in mimicking the speaking style of the input speech query. The overall quality can be improved with a more balanced multi-style TTS dataset and more coherent style labels between the style extraction model training data and the TTS dataset.

8. References

- [1] A. v. d. 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 preprint arXiv:1609.0349*, 2016.
- [2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *3rd International Conference on Learning Representations (ICLR)*, 2015.
- [3] 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 (NeurIPS)*, 2017, pp. 5998–6008.
- [4] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.
- [5] Z. Hodari, O. Watts, S. Ronanki, and S. King, "Learning interpretable control dimensions for speech synthesis by using external data," in *INTERSPEECH*, 2018, pp. 32–36.
- [6] C.-C. Lee, E. Mower, C. Busso, S. Lee, and S. Narayanan, "Emotion recognition using a hierarchical binary decision tree approach," *Speech Communication*, vol. 53, no. 9-10, pp. 1162–1171, 2011.
- [7] E. Mower, M. J. Mataric, and S. Narayanan, "A framework for automatic human emotion classification using emotion profiles," *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 19, no. 5, pp. 1057–1070, 2010.
- [8] K. Han, D. Yu, and I. Tashev, "Speech emotion recognition using deep neural network and extreme learning machine," in *INTERSPEECH*, 2014.
- [9] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5200–5204.
- [10] S. Yoon, S. Byun, and K. Jung, "Multimodal speech emotion recognition using audio and text," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 112–118.
- [11] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, p. 335, 2008.
- [12] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi, "Modeling of various speaking styles and emotions for HMM-based speech synthesis," in *Eighth European Conference on Speech Communication and Technology*, 2003.
- [13] M. Tachibana, J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi, "HMM-based speech synthesis with various speaking styles using model interpolation," in *International Conference on Speech Prosody*, 2004.
- [14] J. Yamagishi, M. Tachibana, T. Masuko, and T. Kobayashi, "Speaking style adaptation using context clustering decision tree for HMM-based speech synthesis," in *2004 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 2004, pp. I–5.
- [15] Y. Wang, R. Skerry-Ryan, Y. Xiao, D. Stanton, J. Shor, E. Battenberg, R. Clark, and R. A. Saurous, "Uncovering latent style factors for expressive speech synthesis," *arXiv preprint arXiv:1711.00520*, 2017.
- [16] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron," *International Conference on Machine Learning (ICML)*, 2018.
- [17] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *International Conference on Machine Learning (ICML)*, 2018.
- [18] F. Locatello, S. Bauer, M. Lucic, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," *International Conference on Machine Learning (ICML)*, 2018.
- [19] N. Tits, F. Wang, K. E. Haddad, V. Pagel, and T. Dutoit, "Visualization and interpretation of latent spaces for controlling expressive speech synthesis through audio analysis," *INTERSPEECH*, 2019.
- [20] G. E. Henter, J. Lorenzo-Trueba, X. Wang, and J. Yamagishi, "Principles for learning controllable TTS from annotated and latent variation," in *INTERSPEECH*, 2017, pp. 3956–3960.
- [21] A. Rabiee, T.-H. Kim, and S.-Y. Lee, "Adjusting pleasure-arousal-dominance for continuous emotional text-to-speech synthesizer," 2019.
- [22] S. King and V. Karaiskos, "The Blizzard challenge 2017," in *Proc. Blizzard Challenge*, vol. 2017, 2017, pp. 1–10.
- [23] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM international conference on Multimedia*, 2013, pp. 835–838.
- [24] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.
- [25] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," *arXiv preprint arXiv:1802.08435*, 2018.
- [26] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognition*, vol. 80, pp. 109–117, 2018.
- [27] R. Banse and K. R. Scherer, "Acoustic profiles in vocal emotion expression," *Journal of personality and social psychology*, vol. 70, no. 3, p. 614, 1996.
- [28] K. R. Scherer, R. Banse, H. G. Wallbott, and T. Goldbeck, "Vocal cues in emotion encoding and decoding," *Motivation and emotion*, vol. 15, no. 2, pp. 123–148, 1991.
- [29] Y. Gao, "Demo for 'interactive text-to-speech via semi-supervised style transfer learning'," <https://github.com/Yolanda-Gao/Interactive-Style-TTS>, 2019, accessed: 2019-10-21.