ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096431

SYNTACC : SYNTHESIZING MULTI-ACCENT SPEECH BY WEIGHT FACTORIZATION

Tuan-Nam Nguyen [1] *Ngoc-Quan Pham* [1] *Alexander Waibel* [1,2]

[1] Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany
[2] Carnegie Mellon University, Pittsburgh, United States

ABSTRACT

Conventional multi-speaker text-to-speech synthesis (TTS) is known to be capable of synthesizing speech for multiple voices, yet it cannot generate speech in different accents. This limitation has motivated us to develop SYNTACC (Synthesizing speech with accents) which adapts conventional multi-speaker TTS to produce multi-accent speech. Our method uses the YourTTS model and involves a novel multi-accent training mechanism. The method works by decomposing each weight matrix into a shared component and an accent-dependent component, with the former being initialized by the pretrained multi-speaker TTS model and the latter being factorized into vectors using rank-1 matrices to reduce the number of training parameters per accent. This weight factorization method proves to be effective in fine-tuning the SYNTACC on multi-accent data sets in a low-resource condition. Our SYNTACC model eventually allows speech synthesis in not only different voices but also in different accents.

Index Terms— Speech synthesis, accent adaptation, multi-speaker TTS, weight factorization, weight decomposition

1. INTRODUCTION

Deep learning approaches have significantly advanced text-to-speech systems in recent years [1][2]. Most TTS systems are trained from a single speaker's voice, but there is contemporary interest in synthesizing voices for any speakers which is known as multi-speaker TTS [3]. Although multi-speaker TTS allows synthesizing speech of any single voice, it cannot produce speech with a specific accent at the same time. Therefore, it could be necessary to use a synthesis system that can handle a variety of accents. In practice, the capability to change an accent is one of the top desired features in TTS systems, particularly by users and communication devices that want to have the TTS voice as their own to communicate with others from different parts of the world. In theory, we also need to generate more non-native accented speech to train augmented speech recognition models [4] [5] or to produce paired audios of the same voice in different accents to train accent conversion models [6]. It is possible for each accent to be processed by a TTS model, only that it requires a lot of training data for each accent. To address this issue, our paper utilizes a multi-accent TTS that can generate many regional accents through a single model. Our proposed system can be fine-tuned from the conventional multi-speaker TTS without a high demand in accented-speech data. The model performance is then evaluated on 4 accents such as Indian accent, Spanish accent, Chinese accent and Vietnamese accent.[1].

Over the years, Transformer-based TTS has become increasingly popular in the research community due to its advantage in long-range context dependencies [7]. Furthermore, Transformer featuring an adaptive weight component is proved to be efficient in various multilingual models, such as for speech recognition, machine translation and speech translation [8]. Hence, our research on SYNTACC aims to contribute to the current literature by investigating the effectiveness of such adaptive component on multi-accent TTS problems. We experiment this method on YourTTS [3], the advanced version of VITS model [9]. As long as matrix-vector multiplication is the primary operation, this method can be applied in any neural architecture and therefore, can be also deployed in an arbitrary TTS neural architecture such as FastSpeech [2] and GlowTTS [1].

2. METHODOLOGY

2.1. Multi-accent adaptive weight component

The idea of adaptive weight is motivated by the belief that there are features shared between accents that must be selectively represented, and networks are expected to switch between different "modes" depending on the input accent being processed. In a multi-accent scheme, the phoneme set, the character set and the word set are the same for every accent. Meanwhile, accents may differ in various aspects such as phonetics (acoustic realisation of the same phoneme), prosody, and pronunciation. From a phonological point of view, accent differences may be reflected in their letter-to-sound mapping [10]. These differences can consist of substitution of some phonemes. For example 'bath' in British English */b a: θ/*

[1]This research was supported in part by a grant from Zoom Video Communications, Inc. The authors gratefully acknowledge the support.

versus in General American English (GA) /b æ θ/; or in insertions/deletions of some phones, for example in 'herb' / ɛː b/ vs /h ɛː b/. Therefore, if we can change the "mode" of the letter-to-sound component of a TTS model, we can expect the accent to alter accordingly. In this paper, the adaptive weight that adds scales and biases to each weight matrix of the letter-to-sound component is selected for investigation.

Adaptive weight factorization is proposed based on the fact that the core operator of neural networks is matrix multiplication [11]. Therefore, it is possible to separate the weight matrix into a shared component W_S and an accent-dependent adaptive scale W_{ML} and bias W_{BL}. The simple matrix multiplication $Y = WX$ becomes:

$$Y = (W_S \cdot W_{ML} + W_{BL})^T X \tag{1}$$

The added weight in this case includes a multiplicative term W_{ML} and a biased term W_{BL}. The term W_{ML} is supposed to directly change the magnitude and direction of the shared weights W_S while the term W_{BL} provides the network with a content-based bias based on the input features X. Each accent retains its distinctive set of W_{ML} and W_{BL}, resulting in a semi-shared architecture.

To encourage the model to share parameters while keeping them efficient, the adaptive weight is factorized as 1-rank matrices that can be represented compactly as a dot-product between two vectors. This factorization can be formed using k vectors per accent, resulting in k independent weight factors, followed by a summation that raises the rank of the additional weight matrices:

$$W = \sum_i^k r_i s_i^T \tag{2}$$

2.2. Adaptive multi-accent speech synthesis

YourTTS serves as the backbone of our SYNTACC which brings in an adaptive component for accents [3]. YourTTS is essentially an improved version of VITS with a number of novel modifications and improvements for zero-shot multi-speaker and multilingual instruction. It is one of the few TTS models that are fully end-to-end, non-autoregressive, and of high-fidelity. Raw text is used as input for YourTTS, in contrast to phonemes being fed to VITS, because this produces more realistic results for many languages, for which there are no grapheme-to-phoneme converters available. The raw text input is also highly suitable for our SYNTACC since the grapheme-to-phoneme for different accents are very different from one another; and in a multi-accent context, it can be challenging to develop grapheme-to-phoneme models in respect to the target accents. The original YourTTS is trained in three languages (English, Portugese and French) but in our experiment, we only focus on English with a multi-accent training setup.

Like YourTTS, our SYNTACC is a conditional variational autoencoder augmented with normalizing flow (Fig 1). In terms of architecture, it basically includes 3 modules: A posterior encoder, a prior decoder and a waveform generator, each encoding the distributions $q_\theta(z|x)$, $p_\psi(z|c)$ and $p_\phi(x|z)$ respectively. $q_\theta(z|x)$ and $p_\psi(z|c)$ are the posterior and data distributions, parameterized by neural posterior encoder's parameters θ and Hifi-GAN [12] waveform generator's paremeter ψ respectively, where x is the speech input and z is the latent variables. The Posterior Encoder receives linear spectrograms and speaker embeddings as input and predicts a latent variable z. This latent variable and speaker embeddings are then used as input to the Hifi-GAN vocoder generator which generates the waveform. The prior of z is defined as $p_\psi(z|c)$, where the latents are conditioned on input texts c, the prior distribution is parameterized by text encoder combined with a normalizing flow decoder f. The Flow-based decoder aims to condition the latent variable z and speaker embeddings with respect to a prior distribution. To align the output of flow-based decoder with the output of the text encoder, we use the Monotonic Alignment Search (MAS). The stochastic duration predictor receives speaker embeddings as inputs and the duration d obtained through MAS. During training, the model aims to maximize the conditional distribution of x given c, denoted as $p(x|c)$ by maximizing its evidence lower bound (ELBO):

$$\log p(x|c) > E_{q_\theta(z|x)}[\log p_\phi(x|z)] - D_{KL}(q_\theta(z|x)||p_\psi(z|c)) \tag{3}$$

As in previous works, we use a text encoder based on a Transformer. For multi-accent training, we combine the embeddings of each input character with 16-dimensional trainable accent embeddings. In addition, we set the number of Transformer blocks to 10 and the number of hidden channels to 196. We can consider it as a baseline model. An adaptive factorized weight with rank 2 is then added to form our proposed model. This has the advantage in directly influencing each layer function, such as the QKV-projection layer in self-attention. The key difference between the baseline model and our proposed model is the text encoder component, since it is considered as a letter-to-sound component. The flow-based decoder comes after the Transformer-based text encoder, which allows an invertible transformation of a simple distribution into a more complex distribution following the rule of change-of-variable. The remaining components, such as a flow-based decoder, Monotonic Alignment Search, a speaker encoder, a posterior encoder, a duration predictor and a waveform generator are inherited from the original paper [3].

3. EXPERIMENTS

3.1. Data and training description

First step, we train YourTTS model with VCTK corpus which contains 44 hours of speech and 109 speakers, sampled at

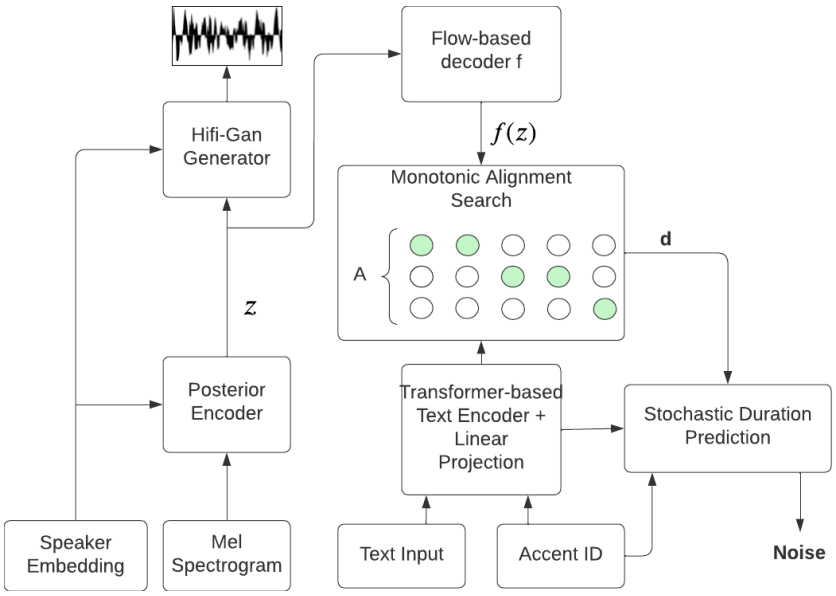

Fig. 1. Multi-accent TTS architecture

48kHz. All audios are sampled at 16kHz before training. The training setting is established in a similar way as in the original implementation [3]. After 150k training steps, this pretrained YourTTS can be used as initialized weights for all experiment models.

Second step, our multi-accent TTS experiments are conducted using L2-Arctic public data set, including Hindi-accented corpus, Spanish-accented corpus, and Vietnamese-accented corpus [13]. These corpuses have 12 speakers in total, thereof 4 native English speakers and 4 Indian accented speakers. Each of them is featured in their respective audios of the same 1152 sentences. To ensure uniformly loud samples and to eliminate extended pauses, pre-processing is performed on all corpuses. Every training audio is normalized to -27dB using the RMS-based normalization from the Python package ffmpeg-normalize and sampled at 16kHz. As a result, we get around 3 hours of Chinese-accented audio, 3 hours of Spanish-accented audio, 3 hours of Indian-accented audio and 3 hours of Vietnamese-accented audio, which can be considered as a low-resource condition.

Third step, we train 4 TTS models - each with a specific accent - and compare their performance with the results of our multi-accent model. To do so, we fine-tune the pretrained YourTTS model in the first step on the accented-speech data processed in the second step. Subsequently, we get 4 single-accent TTS models, namely Indian Accent TTS, Spanish Accent TTS, Chinese Accent TTS and Vietnamese Accent TTS.

The weights of the baseline model and our proposed SYNTACC are both initialized by the pretrained YourTTS. So as not to impair the multi-speaker synthesis capability, we combine the aforementioned accented-speech data with audios of Indian speakers in the VCTK data sets during fine-tuning. To assess how much the pretrained weights contributes to our SYNTACC, we train the SYNTACC model in 3 setups: (1) without pretrained weights; (2) freezing the pretrained weights and fine-tuning only accent-dependent weights; (3) using pretrained weights and fine-tuning all parameters. We use the CoquiAI framework [14] to implement all our experiment models. All models are trained in around 10k steps using an NVIDIA A100 48GB with a batch size of 128.

3.2. Evaluation metrics

We evaluate all experiment models by two types of metrics, objective and subjective, as described in the following section. In the test set, we use 10 sentences, each of which is synthesized in four accents. Sample evaluation audios are available at [2].

3.2.1. Subjective tests

Accentedness, Speaker Similarity Mean Opinion Score (Sim-MOS) and Mean Opinion Score (MOS). For each target accent, three kind of tests are conducted by 10 accented participants who listen to the provided audios and evaluate their overall quality on a 5-point scale: 1-bad, 2-poor, 3-fair, 4-good, 5-excellent. In the Accentedness test, the participants give a score for the degree to which the synthesized audios sound like a specific-accented speech. For the Sim-MOS test, they rate the similarity between the voice identity of the

[2]https://accenttts.github.io/

Models	Accentedness					Sim-MOS					MOS				
	IND	ES	VN	ZH	AVG	IND	ES	VN	ZH	AVG	IND	ES	VN	ZH	AVG
YourTTS specific-accented	**4.7**	3.2	3.8	3.7	3.85	3.7	3.3	3.2	3.7	3.45	4.1	3.7	3.8	3.9	3.88
Baseline	3.2	2.3	2.8	2.9	2.8	4.0	3.9	4.0	3.9	3.95	4.15	4.1	3.9	3.8	3.99
SYNTACC															
+ no pretrained weights	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
+ freeze shared weights	**4.7**	3.1	3.7	4.0	**3.9**	3.65	4.1	3.9	4.3	**3.99**	4.0	3.6	3.8	3.7	3.78
+ fine-tune all	**4.8**	3.5	3.8	3.9	**4.0**	3.9	4.2	3.6	4.0	**3.925**	3.8	3.75	4.2	3.9	3.91

Table 1. Subjective metrics

Models	WER				
	IND	ES	VN	ZH	AVG
YourTTS specific-accented	2.4	3.5	3.7	3.9	3.4
Baseline	1.9	3.2	3.3	3.5	3.0
YourTTS	_	_	_	_	0.8
SYNTACC					
+freeze shared weights	2.1	3.2	3.5	3.8	3.15
+fine-tune all	1.8	3.3	3.1	3.2	**2.85**

Table 2. Objective metrics

output audios and the original audios of target speakers on a scale of 5 as above. By this metric, we want to verify if the multi-speaker synthesis capability remains unimpaired after being fine-tuned. Finally, the MOS test is scored for how natural the output speech sound. In all metrics, the higher rating is the better.

3.2.2. Objective tests

Word Error Rates (WER) We compute a WER for all test audios by using our competitive ASR system [4]. We also compute the WER for the audios generated by the original YourTTS, hence WER can be used as an indicator to see how much an accent influences the output audios. A better multi-accent TTS model is expected to have a lower WER.

3.3. Results

Table 1 and Table 2 present subjective and objective evaluation metrics for Chinese(ZH), Indian (IND), Vietnamese (VN), and Spanish (ES) accents respectively and also the average score (AVG) of all accents. Our SYNTACC is incapable of synthesizing intelligible speech without pretrained weights, meaning that the synthesized audios are not comprehensible. Therefore, we consider our SYNTACC without pretrained weights to be the worst configuration. This can be explained by data insufficiency to train a whole model with accent-dependent components from scratch. As can be seen from Table 1, the four single-accent YourTTS models trained on specific accented data receive a high Accentedness test score but do not perform well on the Sim-MOS test. The reason is that specific-accented YourTTS is fine-tuned on a database of only 4 accented speakers, such limited number can impair multi-speaker synthesis capability. Our SYNTACC with pretrained weights outperform the baseline multi-accent TTS using accent embedding in the Accentedness test (3.9 and 4.0 vs 2.8 on average) while achieving equally good Sim-MOS scores. It implies that our SYNTACC model using adaptive weights might have added more accented features to the synthesized speech than the baseline. Thanks to the advantage of using YourTTS backbone, all models under different settings score well from 3.78 to 3.99 in the MOS test. The average assessments of our SYNTACC with freezing share weights and fine-tuning all parameters do not differ significantly across all three subjective tests. The Accentedness test of Indian accent has significantly higher scores than other accents. Since English is a popular language in India, the pronunciation patterns of Indian speakers are more consistent than those of other accented speakers, which enables our SYNTACC model to learn better on how to speak with an Indian accent. In terms of objective metrics, all experiment models have higher WER than YourTTS. Among which, the best one is SYNTACC with fine-tuning all parameters (2.85 WER). The Indian accented speech has better score than other accent because the training data for speech recognition has more audios from Indian speakers.

4. CONCLUSION

To conclude, this paper has demonstrated our SYNTACC model with weights factorization method. Such kind of method is truly promising thanks to its possibility to be implemented in any TTS neural architecture. In addition, how a YourTTS pretrained weights can help a SYNTACC model to be trained in low-resource conditions is also described. Though we conduct our experiment with a 4-accent TTS model, the number of accents can be further increased simply by enlarging the size of the accent-dependent components before fine-tuning.

As a future perspective, we desire to investigate how to perform accent conversion using a SYNTACC model. This feature is motivated by the fact that YourTTS has been capable of processing both voice conversion and multi-speaker TTS synthesis in a single model, while currently our presented SYNTACC can handle just the multi-accent TTS.

5. REFERENCES

[1] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon, "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020, NIPS'20, Curran Associates Inc.

[2] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. 2019, vol. 32, Curran Associates, Inc.

[3] Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti, "YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone," in *Proceedings of the 39th International Conference on Machine Learning*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, Eds. 17–23 Jul 2022, vol. 162 of *Proceedings of Machine Learning Research*, pp. 2709–2720, PMLR.

[4] Tuan Nam Nguyen, Thai Son Nguyen, Christian Huber, Ngoc-Quan Pham, Thanh-Le Ha, Felix Schneider, and Sebastian Stüker, "KIT's IWSLT 2021 offline speech translation system," in *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, Bangkok, Thailand (online), Aug. 2021, pp. 125–130, Association for Computational Linguistics.

[5] Ngoc-Quan Pham, Felix Schneider, Tuan-Nam Nguyen, Thanh-Le Ha, Thai Son Nguyen, Maximilian Awiszus, Sebastian Stüker, and Alexander Waibel, "KIT's IWSLT 2020 SLT translation system," in *Proceedings of the 17th International Conference on Spoken Language Translation*, Online, July 2020, pp. 55–61, Association for Computational Linguistics.

[6] Tuan Nam Nguyen, Ngoc-Quan Pham, and Alexander Waibel, "Accent Conversion using Pre-trained Model and Synthesized Data from Voice Conversion," in *Proc. Interspeech 2022*, 2022, pp. 2583–2587.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. 2017, vol. 30, Curran Associates, Inc.

[8] Ngoc-Quan Pham, Alexander Waibel, and Jan Niehues, "Adaptive multilingual speech recognition with pre-trained models," in *Proc. Interspeech 2022*, 2022, pp. 3879–3883.

[9] Jaehyeon Kim, Jungil Kong, and Juhee Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 5530–5540, PMLR.

[10] BalaKrishna Kolluru, Vincent Wan, Javier Latorre, Kayoko Yanagisawa, and Mark J. F. Gales, "Generating multiple-accent pronunciations for TTS using joint sequence model interpolation," in *Proc. Interspeech 2014*, 2014, pp. 1273–1277.

[11] Ngoc-Quan Pham, Tuan-Nam Nguyen, Sebastian Stüker, and Alex Waibel, "Efficient Weight Factorization for Multilingual Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 2421–2425.

[12] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, Eds. 2020, vol. 33, pp. 17022–17033, Curran Associates, Inc.

[13] Guanlong Zhao, Sinem Sonsaat, Alif Silpachai, Ivana Lucic, Evgeny Chukharev-Hudilainen, John Levis, and Ricardo Gutierrez-Osuna, "L2-arctic: A non-native english speech corpus," in *Proc. Interspeech 2018*, 2018, pp. 2783–2787.

[14] Eren Gölge, Edresson Casanova, Alexander Korolev, Thomas Werkmeister, WeberJulian, Thorsten Müller, Reuben Morais, Kirian Guiller, Branislav Gerazov, Thorben Hellweg, Ayush Chaurasia, Jörg Thalheim, Neil Stoker, Katsuya Iida, Nicolas Müller, Rishikesh (), Adonis Pujols, Michael Hansen, bgerazov, mittimithai, Agrin Hilmkil, Markus Toman, geneing, Guy Elsmore-Paddock, Martin Weinelt, QP Hou, jyegerlehner, a froghyar, and Anand..., "coqui-ai/tts: v0.2.1," Aug. 2021.