CS Tacotron: End-to-End Chinese-English Code-Switching Speech Synthesis on the LectureDSP Dataset

Ting-Wei Liu, Tao Tu
Speech Lab 531
National Taiwan University
{R07942089,R07922022}@ntu.edu.tw

Abstract

With the wide success of recent machine learning Text-to-speech (TTS) models, promising results on synthesizing realistic speech have proven machine's capability of synthesizing human-like voices. However, little progress has been made in the domain of Chinese-English code-switching text-to-speech synthesis, where machine has to learn to handle both input and output in a multilingual fashion. In this work, we present Code-Switch Tacotron, which is built based on the state-of-the-art end-to-end text-to-speech generative model (Wang et al., 2017). CS-Tacotron is capable of synthesizing code-switching speech conditioned on raw CS text. Given CS text and audio pairs, our model can be trained end-to-end with proper data pre-processing. Furthurmore, we train our model on the LectureDSP dataset, a Chinese-English code-switching lecture-based dataset, which originates from the course Digital Signal Processing (DSP) offered in National Taiwan University (NTU). We present several key implementation techniques to make the Tacotron model perform well on this challenging multilingual speech generation task. In addition, we present a demo system, which demonstrates the Chinese-English code-switching speech synthesis results of our model. CS-Tacotron possess the capability of generating CS speech from CS text, and speaks vividly with the style of LectureDSP's speaker.

1 Introduction

Modern end-to-end TTS systems have many well known advantages when compared to antique TTS pipelines such as statistical parametric TTS systems. In parametric TTS, speech synthesis are achieved through the composition of multiple components: a text processor which extracts various linguistic features, a duration model, an acoustic feature prediction model, and a complex signal-processing-based vocoder. All of which requires significant amount of domain knowledge and complex design, building such systems consumes substantial engineering efforts. Moreover, the combination of these components leads to more error, as they are all trained separately. However an end-to-end TTS can be trained on raw text and audio pairs, in which a single model covers all the components in statistical parametric TTS system's multi-stage model. In this work, we incorporate Tacotron (Wang et al., 2017), an end-to-end generative TTS model based on sequence-to-sequence (Sutskever et al., 2014) with attention (Bahdanau et al., 2014). Tacotron achieves an outstanding performance when trained and tested on monolingual English data, as reported by Wang et al. (2017). The many advantages of an end-to-end model, makes adaptation of such model to a new type of code-switching data tractable. With the success of Tacotron, we further tackle the problem of Chinese-English code-switching speech synthesis.

Reports reveal that people communicates in code-switching languages in everyday life (Nakayama1 et al., 2018), since modern people may not always communicate in monolingual settings, spoken language technologies such as TTS must be developed to handle multilingual input and output. The original TTS transforms highly compressed source (text) into diversely decompressed result (audio). Since the same text can correspond to different pronunciations or styles, this is a particularly difficult learning task for an end-to-end model. In the case of CS TTS, the model has to cope with an even larger output variation for a given input, as two languages are now mixed together. Not only do pronunciations varies between language and language, the input is also entangled with multilingual units. This results in a large-scale inverse mapping problem: mapping highly compact Chinese-English code-switching text to human-like Chinese-English code-switching speech. Liang et al. (2007) attempted to solve the Chinese-English switching language problem and proposed context-dependent HMM state sharing for their code-switched TTS system, which in this work we address the problem with an alternative approach: an end-to-end neural net model instead of HMM. Despite extensive studies on bilingual spoken language technologies, the Chinese-English code-switching case remains a scant research topic up to now, mainly due to the lack of quality and quantity of Chinese-English text audio parallel data.

Combining the two problems of text-to-speech synthesis and code-switching, in this work we present our solution: CS-Tacotron, a sequence-to-sequence based model with attention built upon the successful work of Wang et al. (2017), together with LectureDSP, a diverse and rich Chinese-English code-switching dataset that contains parallel CS text and audio. The model takes Chinese-English mixed text as input, and outputs predicted spectrograms conditioned on the input text. We use the Griffin-Lim Algorithm (Griffin & Lim, 1984) to perform fast waveform synthesis from spectral magnitude sampled on a linear-frequency scale. To handle both Chinese and English text inputs, we use English characters as subword linguistic units for our model's input. Given raw CS text and audio pairs, our data pre-processing pipeline allows the model to be trained completely from random initialization. It does not require further human effort to label phoneme-level alignment, so it can easily scale to using other datasets that contains acoustic data with transcripts. As a result, the presented CS-Tacotron is capable of handling multilingual Chinese-English input and generates human-like code-switching speech accordingly. We claim that it is the implementation details and tricks that makes the migration of monolingual Tacotron (Wang et al., 2017) to multilingual CS-Tacotron successful, we present our key contributions in the following sections.

2 DATASET - LECTUREDSP

Parallel transcripts and speech of code-switching data are extremely rare and valuable, the LectureDSP originates from the Digital Signal Processing (DSP) course offered in National Taiwan University (NTU), which is also available online: `http://speech.ee.ntu.edu.tw/`. The course DSP is offered by professor Lin-Shan Lee from the Department of Electrical Engineering and Department of Computer Science and Information Engineering in NTU. Chinese-English code-switching lecture speech are recorded and segmented into sentences, with proper multilingual transcripts labeled with human annotation, resulting in a well organized parallel text and audio Chinese-English dataset. We take no credit for the collection and organization of the LectureDSP dataset, it was processed by former members of the NTU Speech Lab. Currently this dataset is not available for public release and remains a private collection in the lab.

2.1 DETAILS AND ANALYSIS

LectureDSP is a lecture-based speech dataset, containing over 13700 short audio clips of a single male speaker giving lectures in a university course. The domain of LectureDSP's content covers general greetings, daily conversations, and technological terms in the field of digital speech processing (DSP). A transcription is provided for each audio clip, clips vary in length from 1 to 18 seconds, with an average length of 3.6 seconds, and a total length of approximately 13.7 hours. We calculate the average code-switching frequency throughout the dataset to be 974 switches per hour in average. Where we consider the use of a English vocabulary to be a switch, the total number of switch is 13346 throughout the dataset, and the number of sentences that contains a switch is 6506 (out of a total of 13720). The speech recordings were conducted in an open space (classroom) using a microphone, which introduce considerable noise in speech quality. The quality of training data can be considered as an upper bound of speech generation quality, in this work we also use this dataset to measure the goodness of our model.

2.2 Transcript Pre-processing

The original transcripts in LectureDSP were consists of a mix of English vocabularies and big5 character codes, these 4 digit codes represents Chinese characters in a one to one mapping relation. We use a code-to-character mapping table (`http://ash.jp/code/cn/big5tbl.htm`) to covert the 4 digit codes into Chinese characters. For example, the original transcript ”OKAY [A66E] [A655][A6EC] [A6AD]” will be converted to ”OKAY 好 各位 早”. We process all transcripts and obtain a new set of sentences that consist of a mix of English words and Chinese characters. These sentences are then converted to Chinese pinyin with tone specification using the Python PyPinYin (`https://pypi.org/project/pypinyin/`) toolkit. For example the Chinese character ”中心” will be converted to ”zho1ng xi1n”, this also is a one-to-one mapping. Now we can treat all inputs as a sequence of English characters (letters), we compressed the large vocabulary dimension of Chinese characters into sequence consists of just only 26 English letters (a-z) and 4 numbers (1-5). However, we found that for successful training and generation, we need to distinguish English letter sequences and Chinese pinyin letter sequences with upper case and lower case letters, respectively. As a result, our model handles the following input tokens: 26 lower case English letters (a-z), 5 numbers signifying the tones in Chinese (1-5), 26 upper case English letters (A-Z) and three special tags ([bos], [eos], [spc]) that denote the start and end of sentences, and the spaces between words.

2.3 Speech Pre-processing

Since LectureDSP contains the speech of a single male speaker, and we know that male voice covers a frequency range of 100Hz to 8KHz, and a microphone range (300 Hz-3.4 kHz) has okay intelligibility but the quality of the voice is fairly compromised. Hence we decide cut off the frequencies below 100Hz by a high-pass filter to remove unwanted noise, and smooth out the frequencies ranging from 100Hz to 300Hz. Furthermore, we normalize the volume of the entire dataset to -10 dB for volume consistency. In Figure 1, we show a sample drawn from the dataset, showing the differences between the pre-processed waveform (above) and the original waveform (below). We found that the simple speech pre-processing scheme we used did help improve the quality of LectureDSP, and we train our model on the processed waveform for better generation quality.

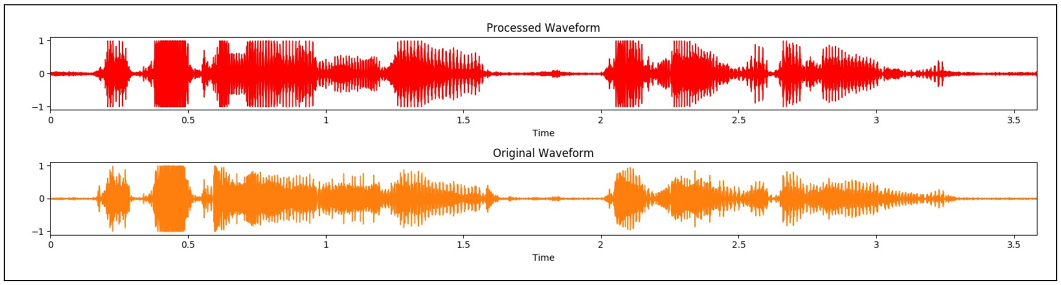

Figure 1: Speech Pre-processing Comparison

3 Model Implementation

The TTS model we used is based on the sequence-to-sequence Tacotron by Wang et al. (2017), where we further modify some architectures and apply several implementation tricks. In this section, we described the main differences of our model from the original work, we believe that these implementation details are crucial to our success. We predict r=5 non-overlapping consecutive output frames at each decoder step instead of r=2, this divides the total number of decoder steps by r, which reduces model size, training time, and inference time. Additionally, we feed all r frames to the next decoder step for the sequence-to-sequence input instead of just the last frame in r frames. The post-processing net’ s task is to convert the seq2seq target (r frames) to linear-scale spectrogram, we scale the loss on these spectrogram so that lower frequencies that corresponds to human speech (0 to 3000 Hz) weighs more. The reason for this modification is that this loss adjustment makes the model attend more on the crucial frequency band of human speech, to be specific we use a weighting of 0.5 * loss(0-3000Hz) + 0.5 * loss(other frequencies) on linear-scale spectrogram loss. We did not use a loss mask in the sequence-to-sequence learning problem, which means we count the loss for all padding, this forces the model to learn when to stop the synthesis sequence. We use a decaying learning rate, as proposed by Vaswani et al. (2017), where the initial learning rate is set to 0.002. Moreover, the CBHG module in Tacotron was modified so that the 1-Dimensional convolution unit has no bias, and we did not disable pre-net's dropout during inference time, we found that the dropout increases the model's generalization for unseen inputs. With the above technological alternation, our model can be successfully trained on CS data. Training time is approximately 24 hours for the model to converge on a GeForce GTX-980 Ti machine.

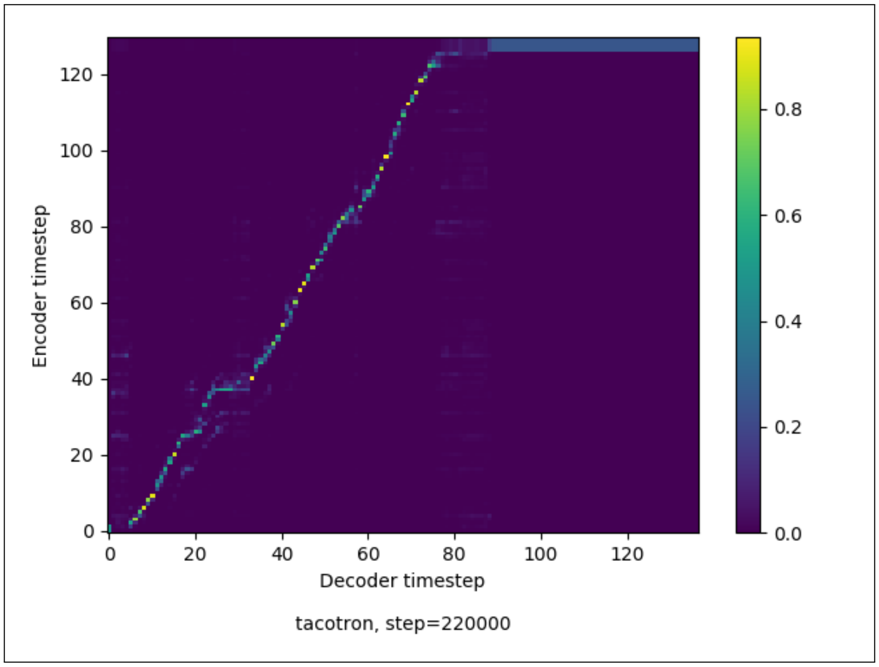

Figure 2: The alignment plot with teacher forcing in training phase.

4 Result

At each timestep, the decoder will attend on parts of the encoder output, i.e., the text embedding , and output r (reduction factor) frames of mel-spectrogram as described by Wang et al. (2017). In Figure 2, we show the alignment plot with teacher forcing of our model in training phase (At the 220000-th iterations). We can see that our model is capable of capturing the monotonic alignment property of the text-to-speech task. We can observe that the alignment plot contains some blurry parts, this is caused by the following two reasons: Firstly, pronouncing consecutive Chinese characters with the same pronunciation is difficult for an end-to-end model. Secondly, the training data (LectureDSP) we used is not the best to suit good TTS generation quality. In Figure 3, we show the alignment plot of our model's testing phase, where the first is the alignment plot of monolingual Chinese input, the second is the alignment plot of Chinese-English code-switching input, and the third is monolingual English input. The inputs are: 這是數位語音處理, 這是數位語音PROCESSING, THIS IS DIGITAL SPEECH PROCESSING, respectively. We can see that our model performs well when performing full Chinese generation and code-switching synthesis, but fails to generate a sentence consists of full monolingual English. We conclude that this is because the training data contains no full English sentences, hence the model fails to generalize from sparse English words contained in Chinese-based code-switching training data. The trained model is capable of synthesizing human-like speech, with a style easily recognizable to be the speaker of the LectureDSP dataset, suggesting that our model is able to capture the style of the trained speaker. We deemed that training is successfully given that the training data is limited and the reference audio's quality is not the best for

TTS training. We present the generation quality of our model in a live demo session, however for the need of reproduction of our results we open source the code we use.

The Pytorch implementation of our work, including the trained model, training algorithms, testing algorithms, and data pre-processing scripts, is available at the following link with a detailed Readme file: `https://github.com/andi611/CS-Tacotron`.

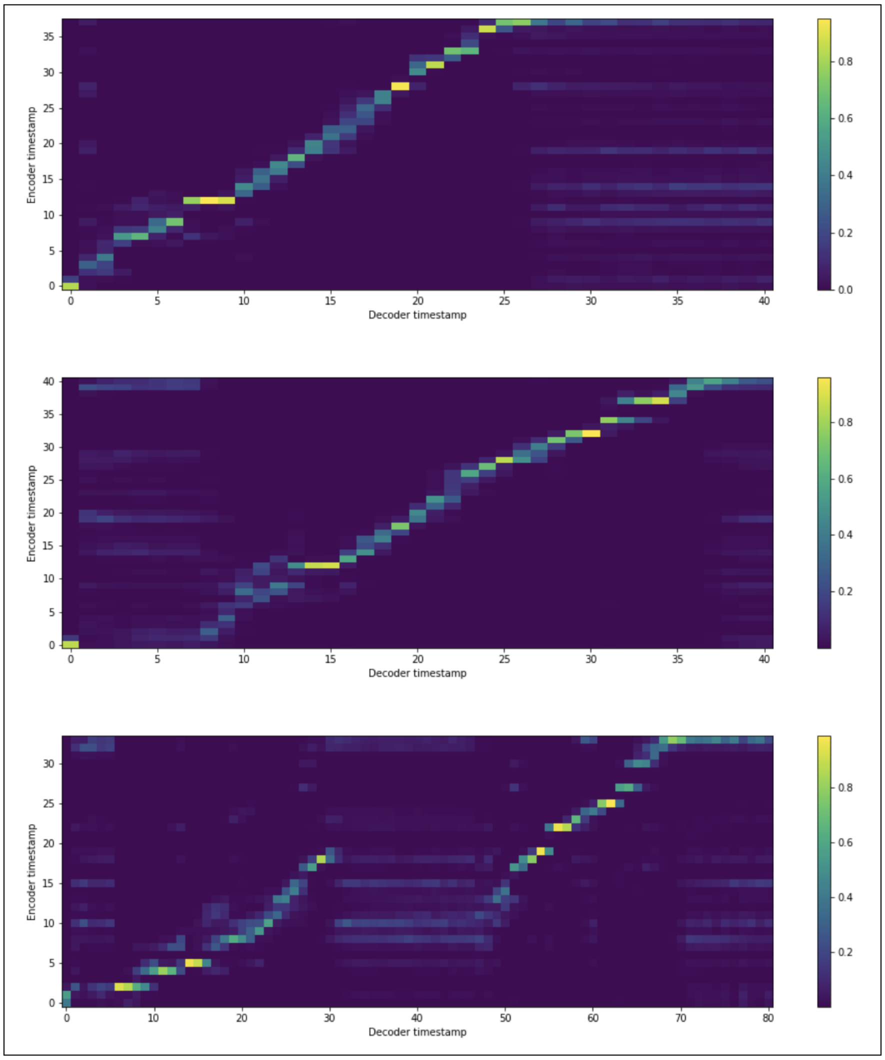

Figure 3: The attention plot without teacher forcing in testing phase. The inputs are: 這是數位語音處理, 這是數位語音PROCESSING, THIS IS DIGITAL SPEECH PROCESSING, respectively.

5 Discussions

We have proposed CS-Tacotron, a code-switching speech synthesis end-to-end generative TTS model based on the work of Wang et al. (2017). Our model takes a character sequence as input, which is provided by our data pre-processing pipeline that transforms Chinese-English mixed word sequences into English character sequence. Given the pre-processed input, our model outputs the corresponding spectrogram, which is later synthesized to code-switching speech. Unlike previous work, our method does not need to handle engineered linguistic features or complex components such as a HMM. It is general and can be easily integrated to other code-switching training schemes, where in this work we prove the training of such TTS model to be successful on our own dataset

LectureDSP. CS-Tacotron successfully modeled the style and code-switching fashion of the speaker, and is capable of generating reasonable and understandable speech utterances.

We have yet to investigate a larger scale model-wise modification to specifically suit the code-switching audio generation paradigm; many designs of the original Tacotron model remains unchanged. The Griffin-Lim-based waveform synthesizer is also well known to produce outputs with audible artifacts, which explains the quality of our generated audio. Future work may include model framework modification, and experimenting with different subword linguistic units for code-switching TTS models.

Acknowledgments

The authors would like to specially thank professor Lin-Shan Lee for offering the wonderful course DSP and providing the LectureDSP dataset. Additionally we thank the members of Speech Lab for constructive discussions and feedback.

Furthermore, we would like to give credit to Ryuichi Yamamoto for this wonderful Pytorch implementation of Tacotron: `https://github.com/r9y9/tacotron_pytorch`, which we mainly based our work on.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236—243, 1984.

Hui Liang, Yao Qian, and Frank K. Soong. Microsoft mulan-a bilingual tts system. *Proc. of ISCA Speech Synthesis Workshop (SSW6)*, pp. 137–142, 2007.

Sahoko Nakayama1, Andros Tjandra1, Sakriani Sakti1, and Satoshi Nakamura1. Speech chain for semi-supervised learning of japanese-english code-switching asr and tts. *IEEE SLT*, 2018.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, pp. 3104—3112, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. arXiv preprint arXiv:1706.03762v5, 2017.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis. arXiv preprint arXiv:1703.10135, 2017.